Atty. Dkt. No.: 068398-0107

U.S. PATENT APPLICATION

for

PARALLEL BLOCK ENCRYPTION METHOD AND MODES FOR DATA CONFIDENTIALITY AND INTEGRITY PROTECTION

Inventors: Virgil Dorin Gligor
Pompiliu Donescu

PARALLEL BLOCK ENCRYPTION METHOD AND MODES FOR DATA CONFIDENTIALITY AND INTEGRITY PROTECTION

CROSS-REFERENCE TO RELATED APPLICATION(S)

[0001] This application claims the benefit of priority under 35 U.S.C Section 119(e) of provisional application serial number 60/227,519 entitled "Fast Parallel XCBC Encryption Modes with Message Integrity" filed on August 24, 2000, the disclosure of which is incorporated herein in its entirety.

FIELD OF THE INVENTION

[0002] The present invention relates to the technical field of secure data communication over insecure channels and secure data storage on insecure media using data encryption techniques. Specifically, the invention relates to encryption methods, program products and systems that achieve both data confidentiality and integrity in a single pass over the data with a single cryptographic primitive, and execute the block-enciphering and deciphering operations necessary for data encryption and decryption in an architecture-independent parallel or pipelined manner.

BACKGROUND OF THE INVENTION

[0003] A long-standing goal in the design of block encryption modes, or schemes, has been the ability to provide both data confidentiality and integrity protection with simple Manipulation Detection Code (MDC) functions, such as the bit-wise exclusive-or, cyclic redundancy code (CRC), or even constant functions (viz., C.M. Campbell: "Design and Specification of Cryptographic Capabilities," in Computer Security and the Data Encryption Standard, (D.K. Brandstad (ed.)) National Bureau of

Standards Special Publications 500-27, U.S. Department of Commerce, February 1978, pp. 54-66; V.D. Gligor and B. G. Lindsay: "Object Migration and Authentication," IEEE Transactions on Software Engineering, SE-5 Vol. 6, November 1979; and R.R. Juneman, S.M. Mathias, and C.H. Meyer: "Message Authentication with Manipulation Detection Codes," Proc. of the IEEE Symp. on Security and Privacy, Oakland, CA., April 1983, pp. 33-54). A desirable property of such modes is that they use only a single pass over the input data with a single cryptographic primitive (i.e., a block cipher) thereby saving processing time and power (viz., V. D. Gligor and P. Donescu's provisional patent application serial number 60/179,147 entitled "XCBC Encryption Schemes" filed on January 31, 2000 and subsequent patent application entitled "Block Encryption Method and Schemes for Data Confidentiality and Integrity Protection."). Another desirable property is that they execute the block-enciphering and deciphering operations necessary for data encryption and decryption in an architecture-independent parallel or pipelined manner.

[0004] Executing block-enciphering and deciphering operations of a mode in an architecture-independent parallel or pipelined manner avoids partitioning the plaintext data into separate segments that can be processed concurrently. The disadvantage of separate encryption, and later decryption, of such segments is that the confidentiality and integrity protection mechanisms must be employed for each segment separately, and this leads to added overhead to processing of the entire plaintext data set. In contrast, the execution of block-enciphering and deciphering operations of a mode in an architecture-independent parallel or pipelined manner implies that the overhead of the confidentiality and integrity protection mechanisms is incurred only once for the entire plaintext data set regardless of how many processing units are used in parallel.

Furthermore, such execution of block-enciphering and deciphering operations has two added advantages, namely (1) the number of processing units need not be known, or negotiated, prior to data encryption or decryption, thereby simplifying the use of the mode in practice, and (2) there is no overhead difference among the parallel, pipelined and sequential execution architecture for data encryption or decryption operations, thereby enlarging the range of the encryption mode applicability in practice.

[0005] A further significant advantage of executing block-enciphering and deciphering operations of a mode in an architecture-independent parallel or pipelined manner is that of efficient incremental and out-of-order processing of such operations; i.e., incremental and out-of-order processing on a per-block basis, as opposed to that on a per-segment basis, has the advantage of lower processing overhead. Incremental processing of block-enciphering and deciphering operations of a mode means that if a small section of a large encrypted message or data set, for instance a single block, is updated, the entire message or data set need not be decrypted, updated, and re-encrypted. Instead, only the blocks affected by the update and that containing the MDC would be decrypted, updated, and re-encrypted. As a result, a substantial performance loss is avoided. Out-of-order processing of block-enciphering and deciphering operations of a mode means that if a block of a message data set arrives at the encryption or decryption processing unit before the blocks preceding it in the message or data set, the processing unit need not wait until all preceding blocks arrive and are processed before processing the block that arrived first. As a consequence, encryption and decryption processing slow-downs are avoided.

[0006] Most attempts to provide both data confidentiality and integrity using only a single processing pass over the input data with a single cryptographic primitive focused on different variations of the Cipher Block Chaining (CBC) mode of encryption (viz., NBS FIPS Pub 81, titled "DES Modes of Operation", National Bureau of Standards, U.S. Department of Commerce, December 1980), which is the most common block-encryption mode in use. However, the CBC mode cannot support parallel or pipelined operation of block-enciphering and deciphering operation in an architecture-independent manner due to the fact that CBC processes each plaintext block sequentially; i.e., the enciphering of each block of a sequence of blocks requires the result of the enciphering of the previous block in the sequence, except the enciphering of the first block in which case the previous block is an initialization vector. Hence parallel or pipelined processing of block enciphering and deciphering operations requires the partitioning of the plaintext data into separate segments that can be processed concurrently.

[0007] The stateful XOR (XORC) mode (viz., M. Bellare, A. Desai, E. Jokipii, and P. Rogaway: "A Concrete Security Treatment of Symmetric Encryption," Proceedings of the 38th Symposium on Foundations of Computer Science, IEEE, 1997, pp. 394-403), which is also known as the "counter-mode," is a well-known mode of encryption whose block-enciphering and deciphering operations can be performed in an architecture-independent parallel or pipelined manner. However, this mode provides only data confidentiality protection but does not provide integrity protection in a single pass using non-cryptographic MDC. The encryption and decryption equations of the stateful XOR (XORC) mode use a counter, ctr, which is initialized to constant value c. Encryption of plaintext string $x = x_1 \ldots x_n$ to obtain ciphertext string $z = z_1 \ldots z_n$ with the XORC mode is defined by the following equation:

$z_i = F_K(ctr+i) \oplus x_i$, $i = 1, \ldots, n$, where the new counter value ctr + n is obtained after each message x encryption, n is the number of blocks of message x, and F_K is the block cipher F using key K. In this mode, decryption of ciphertext string $z = z_1 \ldots z_n$ to obtain plaintext string $x = x_1 \ldots x_n$, is defined by the following equation:

$x_i = F_K(ctr+i) \oplus z_i, i = 1, \ldots, n.$

[0008] It is well-known in the art that the counter (XORC) mode is secure with respect to confidentiality (secrecy) when chosen-plaintext attacks are launched by an adversary using a well-defined set of resources. For example, M. Bellare, A. Desai, E. Jokipii, and P. Rogaway, in "A Concrete Security Treatment of Symmetric Encryption," Proceedings of the 38th Symposium on Foundations of Computer Science, IEEE, 1997, pp. 394-403, demonstrate that the CBC and XOR modes are secure in the left-or-right (or real-or-random) sense, which in turn implies that they are confidentiality-secure against chosen-plaintext attacks (viz., S. Goldwasser and M. Bellare: "Lecture Notes on Cryptography", 1999, available at http://www-cse.ucsd.edu/users/mihir/papers/gb.pdf). In such attacks, an adversary can obtain ciphertexts for a set of plaintexts of his/her own choice. Security with respect to confidentiality (secrecy) means that, after such an attack, the adversary cannot determine the plaintext of a never-seen-before ciphertext message (i.e., a ciphertext message not obtained during the attack) with more than negligible probability. The notion of negligible probability in such attacks is also known to those skilled in the art (e.g., as defined by M. Naor and O. Reingold: "From Unpredictability to Indistinguishability: A Simple Construction of Pseudo-Random Functions from MACs," in Advances in Cryptology - CRYPTO '98 (LNCS 1462), pp. 267-282, 1998). All modes that are secure in this sense are called "confidentiality-secure against chosen-plaintext attacks," or simply, "confidentiality-secure," henceforth.

[0009] It is also well known to those skilled in the art that the counter (XORC) mode does not, by itself, preserve data or message integrity (authenticity), and that non-cryptographic MDC functions cannot be used with counter (XORC) mode to preserve data or message integrity (authenticity). For example, a change of a ciphertext bit position leads to a change in the same bit position of the plaintext and hence simple, efficient MDC functions, such as the bit-wise exclusive-or, cannot be used for integrity protection. Most Message Authentication Code (MAC) modes that can be used to protect the integrity of data or messages encrypted with counter (XORC) mode, such as HMAC (viz., M. Bellare, R. Canetti, and H. Krawczyk, "Keying Hash Functions for Message Authentication," Advances in Cryptology - CRYPTO '96, Springer-Verlag, LNCS 1109, pp. 1-15, 1996), and UMAC (viz., J. Black, S. Halevi, H. Krawczyk, T. Krovetz, and P. Rogaway, "UMAC: Fast Message Authentication via Optimized Universal Hash Functions," Advances in Cryptology - CRYPTO '99, Springer-Verlag, LNCS 1666, 216-233, 1999), cannot operate in an architecture-independent parallel or pipelined manner, thereby decreasing the performance of the added MAC processing pass. Even when MAC modes that operate in an architecture-independent parallel or pipelined manner thereby matching the properties of the counter (XORC) mode, such as the XOR-MAC (viz., M. Bellare, R. Guerin, and P. Rogaway, "XOR-MACs: New Methods for Message Authentication Using Finite Pseudo-Random Functions," Advances in Cryptology – CRYPTO '95, Springer-Verlag, LNCS 963, pp. 15-28; and M. Bellare, R. Guerin, and P. Rogaway, "Method and Apparatus for Data Authentication in a Communication environment," U.S Patent No. 5,757,913, dated 26 May 1998.) and the XECB MAC modes invented by

Gligor and Donescu (viz., V.D. Gligor and P. Donescu's provisional patent application number 60/193,447 entitled "XCBC Encryption Modes and XECB Authentication Modes" filed on March 31, 2000 and subsequent patent application entitled "Authentication Method and Schemes for Data Integrity Protection") are used, the additional MAC processing pass would require substantial added implementation complexity, cost, and power consumption. Thus, such modes would be less suitable for use in low-power applications, and low-power, low-cost hardware devices.

[0010] A well-understood consequence of combining the counter (XORC) mode and a MAC mode for maintaining the integrity (authenticity) of encrypted data or messages is the lack of error recovery for the resulting mode of operation (viz., A.J. Menezes, P.C. van Oorschot, and S.A.Vanstone in their book "Handbook of Applied Cryptography", CRC Press, Boca Raton, 1997, Chapter 7.) That is, any bit error in the ciphertext of an encrypted message or data set whose integrity is protected causes the entire plaintext obtained from ciphertext decryption to be discarded by the mode operation with high probability. Although this is a desirable outcome in all environments where protection against ciphertext forgeries is required, it is sometimes important to enable recovery of the plaintext blocks that are unaffected by errors in the ciphertext block being decrypted. Recovery of plaintext blocks unaffected by ciphertext errors is particularly important in environments of use where retransmission of large-message ciphertext (e.g., video, sound, real-time data streams) following detection of ciphertext errors in a small number of blocks cannot be tolerated by the application whereas some loss of plaintext blocks can be tolerated.

[0011] V. D. Gligor and P. Donescu invented a block encryption method and modes of operation that provide both data confidentiality and integrity with a single cryptographic primitive and a single processing pass over the input plaintext string by using a non-cryptographic MDC function (e.g., bit-wise exclusive-or) for secure data communication over insecure channels and for secure data storage in insecure media (viz V. D. Gligor and P. Donescu's provisional patent application serial number 60/179,147 entitled "XCBC Encryption Schemes" filed on January 31, 2000 and subsequent patent application entitled "Block Encryption Method and Schemes for Data Confidentiality and Integrity Protection," and V. D. Gligor and P. Donescu's provisional patent application serial number 60/193,447 entitled "XCBC Encryption Modes and XECB Authentication Modes" filed on March 31, 2000). The encryption and decryption equations of these modes illustrate in a brief manner how these modes use F_K, a block cipher F with key K, and its inverse F^{-1}_K, to process the plaintext and ciphertext blocks of a message or data. For example, in one of these modes, encryption of plaintext string $x = x_1 \ldots x_n$ to obtain ciphertext string $y = y_1 \ldots y_n$ is defined by the following equations:

$y_i = F_K(x_i \oplus z_{i-1})$ op E_i, $z_0 = F_K(r_0 + 1)$, $x_{n+1} = z_0 \oplus x_1 \oplus \ldots \oplus x_n$, $y_0 = F_K(r_0)$, $i = 1, \ldots, n+1$,

where F_K is the block cipher F using secret key K, r_0 is a secret random number uniformly distributed of the same size as that of a block of the block cipher (i.e., of ℓ bits in length), and E_i is an ℓ-bit element of a sequence of unpredictable elements (e.g., $E_i = i \text{ x } r_0$), and op is a operation that has the inverse op^{-1} (e.g., op can be modulo 2^{ℓ} addition, modulo 2^{ℓ} subtraction, bit-wise exclusive-or). In this mode, decryption of ciphertext string $y = y_0 y_1 \ldots y_{n+1}$ to obtain plaintext string $x = x_1 \ldots x_n$, is defined by the following equations:

$x_i = F^{-1}_K (y_i \; op^{-1} \; E_i) \oplus z_{i-1}$, $i = 1, \ldots, n+1$,

subject to the integrity check $x_{n+1} = z_0 \oplus x_1 \oplus \ldots \oplus x_n$.

[0012] An important added security feature of this mode, which is not shared by other modes, is that the integrity check includes the unpredictable vector z_0. This removes the ability of an adversary to use the integrity check for the purpose of verifying the validity of plaintext x_1, ..., x_n x_{n+1} obtained using guessed keys, as would be the case in typical key-search attacks.

[0013] Gligor and Donescu's block encryption method and modes of operations allow encryption and decryption in parallel or pipelined manners by the segmentation of the plaintext data and of corresponding ciphertext. These modes can also support error recovery at the segment level, since the integrity of each message or data-set segment is separately verified. Thus the recovery of the plaintext segments that are unaffected by errors in the ciphertext being decrypted can be performed by identifying the segments whose integrity checks have passed. Although these modes are suitable for high-performance and low-power applications and for real-time applications, and can be used in low-power, low-cost hardware devices, they cannot support architecture-independent parallel and pipelined operation efficiently at the level of individual block processing.

[0014] Later, C.S. Jutla also designed a block encryption modes of operation that provide both data confidentiality and integrity with a single cryptographic primitive and a single processing pass over the input plaintext string by using a non-cryptographic MDC function (i.e., bit-wise exclusive-or, viz., C.S. Jutla's "Encryption Modes with Almost Free Message Integrity" IBM Thomas Watson Reserch Center, Yorktown Heights, N.Y. 10598, available at *http://eprint.iacr.org/2000/039*, August 2000 version). The encryption and decryption equations of these modes illustrate in a brief manner how these modes use F_K, a block cipher F with key K, and its inverse F^{-1}_K, to process the plaintext and ciphertext blocks of a message or data. For example, in the fastest mode proposed by C.S. Jutla, encryption of plaintext string $x = x_1 \ldots x_n$ to obtain ciphertext string $y = y_1 \ldots y_n$ is defined by the following equations:

$y_i = F_K(x_i \oplus S_i) \oplus S_i$, $x_{n+1} = x_1 \oplus \ldots \oplus x_n$, $y_0 = F_K(r_0)$, $i = 1, \ldots, n+1$,

where F_K is the block cipher F using secret key K, r_0 is a secret random number uniformly distributed of the same size as that of a block of the block cipher (i.e., of ℓ bits in length) generated anew for each message; S_i is an ℓ-bit element of a per-message sequence of random, pairwise-independent elements defined as $S_i = (r_1 + i \times r_2) \bmod p$, where p is a large prime slightly smaller than 2^{ℓ}, $r_1 = F_{K'}(r_0+1)$, $r_2 = F_{K'}(r_0+2)$; and K' is a second key. Two elements S_i and S_j, $i \neq j$ of a sequence of random numbers are pair-wise independent if, for any constants a and b, Probability (S_i = a and S_j = b) = Probability (S_i = a) × Probability (S_i = b). In this mode, decryption of ciphertext string $y = y_0y_1 \ldots y_{n+1}$ to obtain plaintext string $x = x_1 \ldots x_n$, is defined by the following equations:

$x_i = F^{-1}_K (y_i \oplus S_i) \oplus S_i$, $i = 1, \ldots, n+1$,

subject to the integrity check $x_{n+1} = x_1 \oplus \ldots \oplus x_n$.

[0015] The above equations indicate that all inputs to the block-enciphering and deciphering operations (i.e., the inputs of F_K and F^{-1}_K) are independent of the outputs of those operations and hence can be executed in an architecture-independent parallel or pipelined manner. However, Jutla's modes have several performance disadvantages. First, the generation of random, pair-wise independent sequences of elements is less efficient than that of sequences whose elements are only unpredictable, but not pair-wise independent. For example, the computation of the elements of sequence S_i is less efficient than that of sequence E_i used in Gligor and Donescu's modes. This is the case because computation of sequence S_i requires two extra block-enciphering operation (i.e., two operations for r_1 and r_2) per message and modular p additions where p is a prime, which is less efficient than modulo 2^ℓ addition operations. Second, Jutla's modes require that a different sequence S_i be generated for each message and does not allow a single, per-key sequence. This means that these modes can never attain the minimum number of block-enciphering/deciphering operations (i.e., $n+1$ operations for n-block data set) and cannot come close to the minimum latency (i.e., the elapsed time between the beginning and end of message encryption) for parallel operation (i.e., close to the latency of a single block enciphering/deciphering operation) in the processing of a message. For example, Jutla's fastest mode requires $n + 4$ block-cipher invocations, instead of the minimum $n + 1$, for an n-block data set, and a latency of at least three sequential block-cipher invocations regardless of how many parallel processing units are available (i.e., the per-message random number generation, which accounts for at least one block cipher invocation, is followed by the generation of S_i, which accounts for a second block cipher invocation, which is then followed by the parallel invocation on $n + 1$ block cipher operations, which accounts for the latency of a third block cipher invocation). These performance disadvantages are particularly relevant for processing relatively short data sets (e.g., under 256 bytes). Finally, none of Jutla's modes provide any means for message or data set segmentation and have no applicability in environments where recovery from ciphertext errors is required.

[0016] Recently, Katz and Yung proposed a new mode of encryption that uses a single cryptographic primitive and non-cryptographic MDC function to protect confidentiality and integrity called The Related Plaintext Chaining (RPC) (viz., J. Katz and M. Yung, "Unforgeability and Chosen-Ciphertext-Secure Modes of Operation," Proc. of the Fast Software Encryption 2000, B. Schneier (ed.), Springer-Verlag, LNCS). The single processing pass used by this mode is over a modified plaintext that expands the plaintext data by concatenating each plaintext block's identifier with the actual plaintext data of that block to form the input block submitted to block enciphering. (Each block's identifier represents the addition of either a per-message counter or a per-message random number, depending on whether a stateful or stateless mode is desired, and the sequence number of that block in the input data.) Two separate ciphertext blocks are created that represent the enciphering of a message start and end markers. RPC supports architecture-independent parallel and pipelined execution of block enciphering and deciphering operations. However, these operations are over an expanded input plaintext thereby requiring extra block-enciphering and deciphering operations; i.e., up to twice as many as necessary for long messages. Thus the performance and power-consumption characteristics of this mode are inferior to that of single pass modes that do not expand the plaintext input, such as those of Gligor and Donescu and Jutla's that are referred to above. Further, like Jutla's modes, RPC and counter (XORC) mode does not provide any means for message or data set segmentation and have no applicability in environments where recovery from ciphertext errors is required.

SUMMARY OF THE INVENTION

[0017] The inventors have recognized, and it is an aspect of this invention, that it is highly advantageous to provide parallel encryption modes that (1) provide both data confidentiality and integrity and require only one processing pass over the data or message with only one cryptographic primitive (i.e., the block cipher), and (2) perform the block enciphering and deciphering operations in an architecture-independent parallel or pipelined manner without requiring any plaintext expansion, and in a preferred embodiment (3) provide error recovery.

[0018] The inventors have further recognized, and it is an aspect of this invention, that it is advantageous to provide (1) stateless, (2) stateful-sender, and (3) stateful encryption modes, each mode being preferable over the others in different application environments. Many of the prior-art encryption modes provided only stateless modes, which require a high-performance random number generator that produces a new random number for the encryption of each message. Such random number generators may be unavailable or may be hard to protect in terms of confidentiality, integrity and availability; e.g., the new random number used in each message encryption by the sender must be securely transmitted to the receiver, which usually costs at least an additional block-cipher invocation. Other prior-art encryption modes are stateful-sender modes (e.g., a counter-based mode) that eliminate the need for using random number generators, but do not eliminate the extra block-cipher invocation and the need to protect the extra sender-state variables; i.e., the source of randomness is replaced by the enciphering of a message counter, but the counter must be maintained and its integrity must be protected by the sender across encryption of multiple messages, which was unnecessary in stateless modes. It has been further recognized by the present inventors, and is an aspect of this invention that maintaining secret shared-state variables for both the sender and receiver, as opposed to just sender-state, helps eliminate the extra block-cipher invocations, thereby increasing encryption performance, particularly for short messages. However, enlarging the shared state beyond that of a shared secret key may increase the exposure of the mode to physical attacks beyond that possible in stateless and stateful-sender modes. Hence, there remains a need for all three implementation options (i.e., stateless, stateful-sender, and stateful) of an encryption mode.

[0019] Briefly, the present invention comprises, in a first embodiment, a parallel encryption method for providing both data confidentiality and integrity for a message, comprising the steps of: receiving an input plaintext string comprising a message; generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string; creating an MDC block of ℓ bits in length that includes the result of applying a non-cryptographic Manipulation Detection Code (MDC) function to the plurality of the equal-size blocks; presenting the equal-size blocks and the MDC block to a selected parallel encryption mode that makes one and only one processing pass with a single cryptographic primitive over each of the equal-size blocks and the MDC block to create a plurality of hidden ciphertext blocks each of ℓ bits in length; and performing a hidden ciphertext randomization function over the plurality of hidden ciphertext blocks to create a plurality of output ciphertext blocks each of ℓ bits in length.

[0020] In a further aspect of the present invention, the selected parallel encryption mode is confidentiality-secure against chosen-plaintext attacks, wherein each of the equal-size blocks and the MDC block is processed by a block cipher using a secret key (K) to obtain the plurality of hidden ciphertext blocks; and wherein the performing a hidden ciphertext randomization function step comprises combining each of the hidden ciphertext blocks with a corresponding element of a sequence of unpredictable elements for the hidden ciphertext to create a set of output blocks of the ciphertext, wherein a hidden ciphertext block identified by an index i is combined with the element of the sequence identified by index i by an operation for the hidden ciphertext that has an inverse.

[0021] In a further aspect of the present invention, the selected parallel encryption mode that is confidentiality-secure against chosen-plaintext attacks comprises the steps of: performing a plaintext randomization function over the plurality of equal-sized blocks of the plaintext and the MDC block to create a plurality of hidden plaintext blocks each of ℓ bits in length; and processing each of the hidden plaintext blocks by a block cipher using the secret key (K) to obtain the plurality of hidden ciphertext blocks.

[0022] In a further aspect of the present invention, the performing a plaintext randomization function step comprises combining each of the equal-size blocks and the MDC block with a corresponding element of a sequence of unpredictable elements for the hidden plaintext to create a set of hidden plaintext blocks, wherein an equal-size block or the MDC block identified by an index i is combined with the element of the sequence identified by index i by an operation for the hidden plaintext that has an inverse.

[0023] In a further aspect of the present invention, the result of the combination of any two different unpredictable elements of the sequence of unpredictable elements for the hidden ciphertext by the inverse operation of the operation for the hidden ciphertext is unpredictable; and wherein the unpredictable elements selected as the two unpredictable elements for the hidden ciphertext are any two different elements of the same sequence of unpredictable elements for the hidden ciphertext used for the encryption of the plaintext string; and wherein the unpredictable elements selected as the two unpredictable elements for the hidden ciphertext are any two different elements of a plurality of sequences of unpredictable elements for the hidden ciphertext used for encryption of a plurality of plaintext strings with the same secret key K.

[0024] In a further aspect of the present invention, the result of the combination of any two different unpredictable elements of the sequence of unpredictable elements for the hidden plaintext by the inverse operation of the operation for the hidden plaintext is unpredictable; and wherein the unpredictable elements selected as the two unpredictable elements for the hidden plaintext are any two different elements of the same sequence of unpredictable elements for the hidden plaintext used for the encryption of the plaintext string; and wherein the unpredictable elements selected as the two unpredictable elements for the hidden plaintext are any two different elements of a plurality of sequences of unpredictable elements for the hidden plaintext used for encryption of a plurality of plaintext strings with the same secret key K.

[0025] In a further aspect of the present invention, any two different unpredictable elements of the sequence of unpredictable elements for the hidden ciphertext are not pair-wise independent; wherein the unpredictable elements selected as the two unpredictable elements for the hidden ciphertext are any two different elements of the same sequence of unpredictable elements for the hidden ciphertext used for the encryption of the plaintext string; and wherein the unpredictable elements selected as the two unpredictable elements for the hidden ciphertext are any two different elements of a plurality of sequences of unpredictable elements for the hidden ciphertext used for encryption of a plurality of plaintext strings with the same secret key K; and wherein any two different unpredictable elements of the sequence of unpredictable elements for the hidden plaintext are not pair-wise independent; wherein the unpredictable elements selected as the two unpredictable elements for the hidden plaintext are any two different elements of the same sequence of unpredictable elements for the hidden plaintext used for the encryption of the plaintext string; and wherein the unpredictable elements selected as the two unpredictable elements for the hidden plaintext are any two different elements of a plurality of sequences of unpredictable elements for the hidden plaintext used for encryption of a plurality of plaintext strings with the same secret key K.

[0026] In a further aspect of the present invention, the creating an MDC block step comprises applying the non-cryptographic MDC function to the equal-sized blocks of the plaintext.

[0027] In a further aspect of the present invention, the non-cryptographic MDC function is the bit-wise exclusive-or function.

[0028] In a further aspect of the present invention, the non-cryptographic MDC function is the addition modulo $2^{\ell} - 1$ function.

[0029] In a further aspect of the present invention, the non-cryptographic MDC function is the subtraction modulo 2^{ℓ} - 1function.

[0030] In a further aspect of the present invention, there is provided the step of combining the result from applying the non-cryptographic Manipulation Detection Code function to the plurality of equal-sized blocks of the plaintext with a secret, ℓ-bit random vector generated on a per-message basis to obtain the MDC block.

[0031] In a further aspect of the present invention, the combining step comprises performing the combination using a bit-wise exclusive-or function.

[0032] In a further aspect of the present invention, the combining step comprises performing the combination using addition modulo 2^{ℓ} - 1.

[0033] In a further aspect of the present invention, the combining step comprises performing the combination using subtraction modulo 2^{ℓ} - 1.

[0034] In a further aspect of the present invention, there is provided the step of generating the secret random vector from a secret random number generated on a per-message basis.

[0035] In a further aspect of the present invention, there is provided the step of appending the created MDC block after a last block of the set of equal-sized blocks of the plaintext.

[0036] In a further aspect of the present invention, the hidden ciphertext blocks from the processing step comprise n + 1 hidden ciphertext blocks each of ℓ-bit length, where n is the total number of blocks in the set of equal-sized blocks of the plaintext.

[0037] In a further aspect of the present invention, there is provided the step of generating each of a plurality of the unpredictable elements of the sequence of unpredictable elements for the hidden ciphertext by combining a different element identifier for each of the unpredictable elements and a secret random number.

[0038] In a further aspect of the present invention, there is provided the step of generating each of a plurality of the unpredictable elements of the sequence of unpredictable elements for the hidden plaintext by combining a different element identifier for each of the unpredictable elements and a secret random number.

[0039] In a further aspect of the present invention, there are provided the steps of: generating each of a plurality of the unpredictable elements of the sequence of unpredictable elements for the hidden ciphertext by combining a different element identifier for each of the unpredictable elements and a secret random number; and generating each of a plurality of the unpredictable elements of the sequence of unpredictable elements for the hidden plaintext by combining a different element identifier for each of the unpredictable elements and the secret random number.

[0040] In a further aspect of the present invention, the step of generating each element in the sequence of unpredictable elements for the hidden ciphertext comprises a modular 2^{ℓ} multiplication of a different unique element identifier (i) for each element in the sequence of unpredictable elements and the secret random number; and wherein generating each element in the sequence of unpredictable elements for the hidden plaintext comprises a modular 2^{ℓ} multiplication of a different unique element identifier (i) for each element in the sequence of unpredictable elements and the secret random number for all the equal-size blocks of the plaintext and by modular 2^{ℓ} multiplication of $(n+2)$ and the secret random number for the MDC block.

[0041] In a further aspect of the present invention, there are provided the steps of: enciphering the secret random number using the block cipher using the secret key (K); and including this enciphered secret random number (y_0) as one of the output ciphertext blocks.

[0042] In a further aspect of the present invention, the secret random number is provided by a random number generator.

[0043] In a further aspect of the present invention, there is provided the steps of: generating the secret random number by enciphering a count of a counter initialized to a constant, the enciphering being performed with the block cipher using the secret key (K); and incrementing the counter by one on every message encryption.

[0044] In a further aspect of the present invention, the counter is initialized to a constant whose value is the ℓ-bit representation of negative one.

[0045] In a further aspect of the present invention, there is provided the step of initializing the counter to a secret value of ℓ bits in length.

[0046] In a further aspect of the present invention, there is provided the step of outputting the counter value as an output block of the encryption mode.

[0047] In a further aspect of the present invention, there are provided the steps of: deriving a block-index-independent unpredictable element; generating each of a plurality of the unpredictable elements of the sequence of unpredictable elements for the hidden ciphertext by combining the block-index-independent unpredictable element with each of a plurality of block-index-dependent unpredictable elements for the hidden ciphertext; and generating each of a plurality of the unpredictable elements of the sequence of unpredictable elements for the hidden plaintext by combining the block-index-independent unpredictable element with each of a plurality of block-index-dependent unpredictable elements for the hidden ciphertext.

[0048] In a further aspect of the present invention, there are provided the steps of: wherein the block-index-independent unpredictable element is obtained from a count of an ℓ -bit counter initialized to a non-zero constant, and a per-key secret, first random initial number shared between sender and receiver; and wherein each of the plurality of block-index-dependent unpredictable elements for the hidden ciphertext is obtained from an ℓ -bit element index and a secret, second random initial number shared between sender and receiver; wherein each of the plurality of block-index-dependent unpredictable elements for the hidden plaintext is obtained from an ℓ -bit element index and a per-key secret, second random initial number shared between sender and receiver; wherein the secret, first and second random initial numbers are independent; and wherein the ℓ -bit counter is incremented by one on every message encryption.

[0049] In a further aspect of the present invention, the combining to obtain the unpredictable elements for the hidden ciphertext comprises an addition modulo 2^{ℓ}.

[0050] In a further aspect of the present invention, the combining to obtain the unpredictable elements for the hidden plaintext comprises an addition modulo 2^{ℓ}.

[0051] In a further aspect of the present invention, the combining to obtain the unpredictable elements for the hidden ciphertext comprises a subtraction modulo 2^{ℓ}.

[0052] In a further aspect of the present invention, the combining to obtain the unpredictable elements for the hidden plaintext comprises a subtraction modulo 2^{ℓ}.

[0053] In a further aspect of the present invention, the combining to obtain the unpredictable elements for the hidden ciphertext comprises a bit-wise exclusive-or operation.

[0054] In a further aspect of the present invention, the combining to obtain the unpredictable elements for the hidden plaintext comprises a bit-wise exclusive-or operation.

[0055] In a further aspect of the present invention, there are provided the steps of: wherein the block-index-independent unpredictable element is obtained by multiplication modulo 2^{ℓ} of the secret, first random initial number with a different value of the counter; and wherein each of the plurality of block-index-dependent unpredictable elements for the hidden ciphertext is obtained by multiplication modulo 2^{ℓ} of the secret, second random initial number with the index i of the hidden ciphertext block; and wherein each of the plurality of block-index-dependent unpredictable elements for the hidden plaintext is obtained by multiplication modulo 2^{ℓ} of the secret, second random initial number with the index i of the plaintext block; and wherein the unpredictable element for the hidden plaintext corresponding to the MDC block is the block-index-independent unpredictable element itself.

[0056] In a further aspect of the present invention, the operation for the hidden ciphertext that has an inverse is the addition modulo 2^{ℓ}.

[0057] In a further aspect of the present invention, the operation for the hidden ciphertext that has an inverse is a bit-wise exclusive-or operation.

[0058] In a further aspect of the present invention, the operation for the hidden ciphertext that has an inverse is the subtraction modulo 2^{ℓ} operation.

[0059] In a further aspect of the present invention, the operation for the hidden plaintext that has an inverse is the addition modulo 2^{ℓ}.

[0060] In a further aspect of the present invention, the operation for the hidden plaintext that has an inverse is a bit-wise exclusive-or operation.

[0061] In a further aspect of the present invention, the operation for the hidden plaintext that has an inverse is the subtraction modulo 2^{ℓ} operation.

[0062] In a further aspect of the present invention, the step of generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string further comprises the steps of: padding the input plaintext string as necessary such that its length is a multiple of ℓ bits; and partitioning the padded input plaintext string into a plurality of equal-size blocks of ℓ bits in length.

[0063] In a further aspect of the present invention, the padding of the input plaintext string is a standard padding method.

[0064] In a further aspect of the present invention, the padding of the input plaintext string step comprises the steps of: if the last block of the plaintext has ℓ bits in length derive a last element of the sequence of unpredictable elements for the hidden plaintext to be combined with the MDC block to form a hidden plaintext block from the bit-wise complement of a random number; else, append to the last block of the plaintext the bit 1 and the necessary bits of 0 to generate a last equal-size block, and derive a last element of the sequence of unpredictable elements for the hidden plaintext to be combined with the MDC block to form a hidden plaintext block from the random number; and generating each but the last of a plurality of the unpredictable elements of the sequence of unpredictable elements for the hidden plaintext by combining a different element identifier for each of the unpredictable elements and the secret random number.

[0065] In a further aspect of the present invention, the padding of the input plaintext string step comprises the steps of: if the last block of the plaintext has ℓ bits in length derive a last element of the sequence of unpredictable elements for the hidden plaintext to be combined with the MDC block to form a hidden plaintext block from a different block-index-independent unpredictable element obtained from the bit-wise complement of a first random number shared between a sender and a receiver; else, append to the last block of the plaintext the bit 1 and the necessary bits of 0 to generate a last equal-size block, and derive the last element of the sequence of unpredictable elements for the hidden plaintext to be combined with the MDC block to form a hidden plaintext block from a different block-index-independent unpredictable element obtained from the first random number shared between a sender and a receiver; and generating each but the last of a plurality of the unpredictable elements of the sequence of unpredictable elements for the hidden plaintext by combining a different block-index-independent unpredictable element obtained from the first random number shared between a sender and a receiver and each of a plurality of block-index-dependent unpredictable elements for the hidden plaintext.

[0066] In a further embodiment of the present invention, there is provided a parallel decryption method that is the inverse of the parallel encryption method which provides both data confidentiality and integrity, comprising the steps of: presenting a string including ciphertext string for decryption; partitioning the ciphertext string into a plurality of ciphertext blocks comprising ℓ bits each; selecting $n+1$ ciphertext blocks from the plurality of ciphertext blocks representing n data blocks and one MDC block and performing a reverse hidden ciphertext randomization function on each of the selected $n+1$ ciphertext blocks to obtain a plurality of hidden ciphertext blocks each of ℓ bits in length; presenting the hidden ciphertext blocks to a selected parallel decryption mode that makes one and only one processing pass with a single cryptographic primitive that is the inverse of an encryption single cryptographic primitive over the plurality of hidden ciphertext blocks to obtain a plurality of plaintext blocks and one decrypted MDC block each of ℓ bits in length; verifying integrity of the plaintext blocks using a non-cryptographic Manipulation Detection Function (MDC) function; outputting the plurality of plaintext blocks as an accurate plaintext string if the integrity verification passes; and outputting a failure indicator if the integrity verification fails.

[0067] In a further aspect of the present invention, the performing the reverse hidden-ciphertext randomization function comprises the steps of: generating a sequence of unpredictable elements for the hidden ciphertext each of ℓ-bit length in the same manner as used at an encryption method; selecting $n+1$ ciphertext blocks from the plurality of ciphertext blocks representing n data blocks and one MDC block in the same order as that used at an encryption method, and combining the selected ciphertext blocks with the sequence of unpredictable elements for the hidden ciphertext to obtain a plurality of hidden ciphertext blocks (z_i), such that each of the n + 1 ciphertext blocks identified by index i is combined with the element of the sequence of unpredictable elements for the hidden ciphertext identified by index i, by the inverse of the operation for the hidden ciphertext used at the encryption method; and wherein the verifying integrity step comprises creating an MDC decryption block by applying the non-cryptographic Manipulation Detection Code function to the n decrypted plaintext data blocks; and comparing the created MDC decryption block with the decrypted MDC block.

[0068] In a further aspect of the present invention, the creating an MDC decryption block further comprises combining the result with a secret, ℓ-bit random vector, the combining operation being the same as the combining operation at the encryption method, and the secret random vector being derived from the secret random number in the same manner as at the encryption method.

[0069] In a further aspect of the present invention, the selected parallel decryption mode comprises the steps of: processing each of the hidden ciphertext blocks with the inverse of the block cipher used at an encryption method using a secret key (K) to obtain a plurality of hidden plaintext blocks; and performing a reverse plaintext randomization function over the plurality of hidden plaintext blocks to create a plurality of n decrypted plaintext data blocks and one decrypted MDC block each of ℓ bits in length.

[0070] In a further aspect of the present invention, performing the reverse plaintext randomization function comprises the steps of: generating a sequence of unpredictable elements for the hidden plaintext each of ℓ-bit length in the same manner as used at an encryption method; and combining the selected hidden plaintext blocks with the sequence of unpredictable elements for the hidden plaintext to obtain a plurality of n plaintext blocks and one decrypted MDC block, such that each of the n+1 hidden plaintext blocks identified by index i is combined with the element of the sequence of unpredictable elements for the hidden plaintext identified by index i, by the inverse of the operation for the hidden plaintext used at the encryption method.

[0071] In a further aspect of the present invention, there are provided the steps of: deriving a secret random number from the ciphertext string presented for decryption; and generating each of a plurality of the unpredictable elements of the sequence of unpredictable elements for the hidden ciphertext in the same manner as at the encryption method.

[0072] In a further aspect of the present invention, there are provided the steps of: deriving a secret random number from the ciphertext string presented for decryption; and generating each of a plurality of the unpredictable elements of the sequence of unpredictable elements for the hidden plaintext in the same manner as at the encryption method.

[0073] In a further aspect of the present invention, there are provided the steps of: deriving a secret random number from the ciphertext string presented for decryption; generating each of a plurality of the unpredictable elements of the sequence of unpredictable elements for the hidden ciphertext in the same manner as at the encryption method; and generating each of a plurality of the unpredictable elements of the sequence of unpredictable elements for the hidden plaintext in the same manner as at the encryption method.

[0074] In a further aspect of the present invention, there are provided the steps of: selecting the ciphertext block of a secret random number (y_0) from the string presented for decryption; and deciphering the selected ciphertext block to obtain the secret random number.

[0075] In a further aspect of the present invention, the deciphering step comprises performing the deciphering with the inverse of the block cipher using the secret key (K).

[0076] In a further aspect of the present invention, there are provided the steps of: for the encryption method generating a secret random number by enciphering a count of a counter initialized to a constant, the enciphering being performed with the block cipher using the secret key; and incrementing the counter by one on every message encryption; and further comprises for decrypting the ciphertext blocks of the partitioned ciphertext string the steps of: selecting a counter block representing the count of the counter from the string presented at decryption; and enciphering the selected counter block to obtain the secret random number.

[0077] In a further aspect of the present invention, the enciphering step comprises performing the enciphering with the block cipher using the secret key.

[0078] In a further aspect of the present invention, there are provided the steps of: generating each of a plurality of the unpredictable elements of the sequence of unpredictable elements for the hidden ciphertext by combining a different block-index-independent unpredictable element with each of a plurality of block-index-dependent unpredictable elements for the hidden ciphertext in the same manner as at the encryption method; and generating each of a plurality of the unpredictable elements of the sequence of unpredictable elements for the hidden plaintext by combining a different block-index-independent unpredictable element with each of a plurality of block-index-dependent unpredictable elements for the hidden plaintext in the same manner as at the encryption method.

[0079] In a further aspect of the present invention, the string presented for decryption is obtained by applying the encryption method that provides both data confidentiality and integrity to an input plaintext string, and further comprises outputting the input plaintext string.

[0080] In yet a further embodiment of the present invention, there are provided a method for segmented encryption processing of a message comprising the steps of: partitioning the input plaintext string into a plurality of input plaintext segments; concurrently presenting each different one of the plurality of input plaintext segments to a different one of a plurality of parallel encryption methods, each of the different methods using a different ℓ-bit secret random number per segment to obtain a ciphertext segment, wherein each encryption method provides both data confidentiality and integrity with a single processing pass over the input plaintext segment and a single cryptographic primitive, and uses a non-cryptographic Manipulation Detection Code function, wherein the single cryptographic primitive is an ℓ-bit block cipher using a secret key; assembling the plurality of ciphertext segments into a ciphertext string; and outputting the ciphertext string.

[0081] In a further aspect of the present invention, the assembling step comprises including in the ciphertext string the number of ciphertext segments, a ciphertext segment index, a length of each ciphertext segment and a sequence of ciphertext segments.

[0082] In a further aspect of the present invention, there is provided the step of: generating the different ℓ-bit secret random number per segment from a secret random number of ℓ bits in length.

[0083] In a further aspect of the present invention, there is provided the step of: generating the different secret random number per segment from the secret random number of ℓ bits by adding modulo 2^{ℓ} a plaintext segment sequence index for that segment to the secret random number.

[0084] In a further aspect of the present invention, there are provided the steps of: generating the secret random number of ℓ bits in length by a random number generator; enciphering the secret random number with the block cipher using a first key (K); and including the enciphered secret random number as an output block of the output ciphertext string.

[0085] In a further aspect of the present invention, there are provided the steps of: generating each of the secret random number per segment by enciphering the result of adding the segment number to a counter initialized to a constant, the enciphering being done with the block cipher using the first key (K); and outputting the counter value as an output block of the output ciphertext string; and incrementing after every different message encryption the counter by a number equal to a number of plaintext segments in the message.

[0086] In a further aspect of the present invention, there are provided the steps of: generating each of the secret random number per segment from a per-key secret, first random initial number shared between sender and receiver and the result of adding modulo 2^{ℓ} the segment number to a counter initialized to a constant, and outputting the counter value as an output block of the output ciphertext string; and incrementing after every different message encryption the counter by a number equal to a number of plaintext segments in the message.

[0087] In a further aspect of the present invention, the generating each of the secret random number per segment comprises multiplying modulo 2^{ℓ} the per-key secret, first random initial number shared between sender and receiver with the result of adding the segment number to the counter.

[0088] In a yet further embodiment of the present invention, there is provided a method for segmented decryption processing of a message comprising the steps of: presenting a string including the ciphertext string of a message for decryption; partitioning the ciphertext string into a plurality of ciphertext segments; concurrently presenting the plurality of ciphertext segments to a plurality of decryption modes; obtaining a different secret random number per ciphertext segment in the same manner as at the segmented encryption method; decrypting each ciphertext segment using the different secret random number per ciphertext segment to obtain a plaintext segment, using a parallel decryption method that is the inverse of the parallel encryption method that provides both data confidentiality and integrity with a single processing pass over the input plaintext segment and a single cryptographic primitive, wherein the single cryptographic primitive is an ℓ-bit block cipher using a secret key, and using a non-cryptographic Manipulation Detection Code function for verifying integrity of the plaintext blocks of each plaintext segment; and verifying the integrity of each plaintext segment and for each plaintext segment, outputting either the plaintext segment if the integrity verification passes, or an error indicator.

[0089] In a further aspect of the present invention, each of the different secret random numbers per ciphertext segment are obtained from a secret random number in the same manner at as used at a segmented encryption method.

[0090] In a further aspect of the present invention, there are provided the steps of: selecting a ciphertext block of the secret random number from the string presented for decryption; and deciphering the selected ciphertext block to obtain the secret random number.

[0091] In a further aspect of the present invention, the method includes performing the deciphering step with the inverse of a block cipher using the secret key, the block cipher and the secret key being the same as to those used at a segmented encryption method.

[0092] In a further aspect of the present invention, there are provided the steps of: for the segmented encryption method generating the secret random number per ciphertext segment by enciphering the result of adding modulo 2^{ℓ} the segment number with a counter initialized to a constant, the enciphering being done with the block cipher using the first key (K); and incrementing after every different message encryption the counter by a number equal to a number of plaintext segments in the message; and further comprising for segmented decryption of the ciphertext segments of the partitioned ciphertext string the steps of: selecting a counter block holding the count of the counter from the string presented for decryption; enciphering the result of adding modulo 2^{ℓ} the segment number with the selected counter block to obtain the secret random number per ciphertext segment.

[0093] In a further aspect of the present invention, the enciphering of the result of adding modulo 2^{ℓ} the segment number with a counter initialized to a constant step comprises enciphering with the block cipher using the same key as that used for segmented encryption.

[0094] In a further aspect of the present invention, there are provided the steps of: for the segmented encryption method generating each of the secret random number per segment from a per-key secret, first random initial number shared between sender and receiver and the result of adding modulo 2^{ℓ} the segment number to a counter initialized to a constant; and outputting the counter value as an output block of the output ciphertext string; and incrementing after every different message encryption the counter by a number equal to a number of plaintext segments in the message; and further comprising for segmented decryption of the ciphertext segments of the partitioned ciphertext string the steps of: selecting a counter block holding the count of the counter from the string presented for decryption; and generating each of the secret random number per ciphertext segment from the per-key secret, first random initial number shared between sender and receiver and the result of adding modulo 2^{ℓ} the segment number to the counter.

[0095] In a yet further embodiment of the present invention, there is provided a parallel encryption method for providing both data confidentiality and integrity for a message, that updates a ciphertext string incrementally, comprising the steps of: receiving an input plaintext string comprising a message; generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string; creating an MDC block of ℓ bits in length that includes the result of applying a non-cryptographic Manipulation Detection Code (MDC) function to the plurality of the equal-size blocks; performing a plaintext randomization function over the plurality of equal-sized blocks of the plaintext and the MDC block to create a plurality of hidden plaintext blocks each of ℓ bits in length; processing each of the hidden plaintext blocks by a block cipher using a secret key (K) to obtain a plurality of hidden ciphertext blocks; performing a hidden ciphertext randomization function over the plurality of hidden ciphertext blocks to create a plurality of output ciphertext blocks each of ℓ bits in length; and further comprising the steps of: receiving an input plaintext string; generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string; receiving an input ciphertext string including a plurality of $n+1$ equal-size blocks of the ciphertext of ℓ bits in length, wherein the $n+1$ block of the ciphertext corresponds to an MDC block for the plaintext string; receiving a new ℓ-bit plaintext block to replace an ℓ-bit plaintext block at index i; creating a new MDC block of ℓ bits in length that includes the result of applying a non-cryptographic

Manipulation Detection Code (MDC) function to the plurality of the equal-size blocks and the new ℓ-bit plaintext block; performing the same plaintext randomization function as that used at a parallel encryption method over the new ℓ-bit plaintext block and the new MDC block to create two new hidden plaintext blocks each of ℓ bits in length using index i for the new ℓ-bit plaintext block and index n+1 for the new MDC block; processing each of the two new hidden plaintext blocks by a block cipher using the secret key (K) to obtain two new hidden ciphertext blocks; performing the same hidden ciphertext randomization function as that used at a parallel encryption method over the two new hidden ciphertext blocks to create two new output ciphertext blocks each of ℓ bits in length using index i for the new ℓ-bit plaintext block and index n+1 for the new MDC block; replacing in the input ciphertext string, the input ciphertext block at index i with the output ciphertext block for the new ℓ-bit plaintext block and replace the input ciphertext block at index n+1 with the output ciphertext block for the new MDC block, to create a new ciphertext string; and outputting the new ciphertext string.

[0096] In a further aspect of the present invention, the generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string further comprises the steps of: padding the input plaintext string as necessary such that its length is a multiple of ℓ bits; and partitioning the padded input plaintext string into a plurality of equal-size blocks of ℓ bits in length.

[0097] In a further aspect of the present invention, there are provided the steps of: receiving a plurality of new ℓ-bit plaintext blocks to replace a plurality of ℓ-bit plaintext blocks at the plaintext string at index i; and providing a parallel encryption method that outputs a ciphertext string incrementally for each of the plurality of new ℓ-bit plaintext blocks.

[0098] In a yet further embodiment of the present invention, there is provided a parallel encryption method for providing both data confidentiality and integrity for a message, comprising the steps of: receiving an input plaintext string comprising a message; generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string; partitioning the padded input plaintext string into a plurality of equal-size blocks of ℓ bits in length; creating an MDC block of ℓ bits in length that includes the result of applying a non-cryptographic Manipulation Detection Code (MDC) function to the plurality of the equal-size blocks; performing a plaintext randomization function over the plurality of equal-sized blocks of the plaintext and the MDC block using a different plaintext index for each equal-sized block and the MDC block to create a plurality of hidden plaintext blocks each of ℓ bits in length; processing each of the hidden plaintext blocks by a block cipher using a secret key (K) to obtain a plurality of hidden ciphertext blocks; performing a hidden ciphertext randomization function over the plurality of hidden ciphertext blocks using a different ciphertext index for each hidden ciphertext block to create a plurality of output ciphertext blocks each of ℓ bits in length; and further providing an out-of-order decryption method for the parallel encryption method, which provides both data confidentiality and integrity, comprising the steps of: receiving a string including a plurality of $n+1$ ℓ-bit ciphertext blocks for decryption; selecting $n+1$ ciphertext blocks from the plurality of ciphertext blocks representing n data blocks and one MDC block and performing a reverse hidden ciphertext randomization function on each of the selected $n+1$ ciphertext blocks using the ciphertext index to obtain a plurality of hidden ciphertext blocks each of ℓ bits in length; processing each of the hidden ciphertext blocks with the inverse of the block cipher used at an encryption method using the secret key (K) to obtain a plurality of hidden plaintext blocks;

and performing an inverse plaintext randomization function over the plurality of hidden plaintext blocks using the plaintext index to create a plurality of n decrypted plaintext data blocks and one decrypted MDC block each of ℓ-bit length; creating an MDC decryption block by applying the non-cryptographic Manipulation Detection Code function to the n decrypted plaintext data blocks in the same manner as at a parallel encryption method; verifying integrity of the plaintext blocks by comparing the created MDC decryption block with the decrypted MDC block; outputting the plurality of plaintext blocks as an accurate plaintext string if the integrity verification passes; and outputting a failure indicator if the integrity verification fails.

[0099] In a further aspect of the present invention, the generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string further comprises the steps of: padding the input plaintext string as necessary such that its length is a multiple of ℓ bits; and partitioning the padded input plaintext string into a plurality of equal-size blocks of ℓ bits in length.

[0100] In a yet further embodiment of the present invention, there is provided a program product for parallel encryption for providing both data confidentiality and integrity for a message, including machine-readable program code for causing a machine to perform the following method steps: receiving an input plaintext string comprising a message; generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string; creating an MDC block of ℓ bits in length that includes the result of applying a non-cryptographic Manipulation Detection Code (MDC) function to the plurality of the equal-size blocks; presenting the equal-size blocks and the MDC block to a selected parallel encryption mode that makes one and only one processing pass with a single cryptographic primitive over each of the equal-size blocks and the MDC block to create a plurality of hidden ciphertext blocks each of ℓ bits in length; and performing a hidden ciphertext randomization function over the plurality of hidden ciphertext blocks to create a plurality of output ciphertext blocks each of ℓ bits in length.

[0101] In a further aspect of the present invention, the program code includes code to cause: the step of presenting the equal-size blocks and the MDC block to a selected parallel encryption mode processing each of the equal-size blocks and the MDC block by a parallel encryption mode to be confidentiality-secure against chosen-plaintext attacks, wherein each of the equal-size blocks and the MDC block is processed by a block cipher using a secret key (K) to obtain the plurality of hidden ciphertext blocks; and to cause the step of performing a hidden ciphertext randomization function step comprises code for combining each of the hidden ciphertext blocks with a corresponding element of a sequence of unpredictable elements for the hidden ciphertext to create a set of output blocks of the ciphertext, wherein a hidden ciphertext block identified by an index i is combined with the element of the sequence identified by index i by an operation for the hidden ciphertext that has an inverse.

[0102] In a further aspect of the present invention, the program code for causing the performance of the step of processing each of the equal-size blocks and the MDC block by a parallel encryption mode that is confidentiality-secure against chosen-plaintext attacks comprises code for: performing a plaintext randomization function over the plurality of equal-sized blocks of the plaintext and the MDC block to create a plurality of hidden plaintext blocks each of ℓ bits in length; and processing each of the hidden plaintext blocks by a block cipher using the secret key (K) to obtain the plurality of hidden ciphertext blocks.

[0103] In a further aspect of the present invention, the program code for performing a plaintext randomization function step comprises code for combining each of the equal-size blocks and the MDC block with a corresponding element of a sequence of unpredictable elements for the hidden plaintext to create a set of hidden plaintext blocks, wherein an equal-size block or the MDC block identified by an index i is combined with the element of the sequence identified by index i by an operation for the hidden plaintext that has an inverse.

[0104] In a further aspect of the present invention, the result of the combination of any two different unpredictable elements of the sequence of unpredictable elements for the hidden ciphertext by the inverse operation of the operation for the hidden ciphertext is unpredictable; and wherein the unpredictable elements selected as the two unpredictable elements for the hidden ciphertext are any two different elements of the same sequence of unpredictable elements for the hidden ciphertext used for the encryption of the plaintext string; and wherein the unpredictable elements selected as the two unpredictable elements for the hidden ciphertext are any two different elements of a plurality of sequences of unpredictable elements for the hidden ciphertext used for encryption of a plurality of plaintext strings with the same secret key K.

[0105] In a further aspect of the present invention, the result of the combination of any two different unpredictable elements of the sequence of unpredictable elements for the hidden plaintext by the inverse operation of the operation for the hidden plaintext is unpredictable; and wherein the unpredictable elements selected as the two unpredictable elements for the hidden plaintext are any two different elements of the same sequence of unpredictable elements for the hidden plaintext used for the encryption of the plaintext string; and wherein the unpredictable elements selected as the two unpredictable elements for the hidden plaintext are any two different elements of a plurality of sequences of unpredictable elements for the hidden plaintext used for encryption of a plurality of plaintext strings with the same secret key K.

[0106] In a further embodiment of the present invention, a program product is provided for parallel decryption that is the inverse of a program product for parallel encryption which provides both data confidentiality and integrity, comprising machine-readable program code for causing a machine to perform the following method steps: presenting a string including ciphertext string for decryption; partitioning the ciphertext string into a plurality of ciphertext blocks comprising ℓ bits each; selecting $n+1$ ciphertext blocks from the plurality of ciphertext blocks representing n data blocks and one MDC block and performing a reverse hidden ciphertext randomization function on each of the selected $n+1$ ciphertext blocks to obtain a plurality of hidden ciphertext blocks each of ℓ bits in length; presenting the hidden ciphertext blocks to a selected parallel decryption mode that makes one and only one processing pass with a single cryptographic primitive that is the inverse of an encryption single cryptographic primitive over the plurality of hidden ciphertext blocks to obtain a plurality of plaintext blocks and one decrypted MDC block each of ℓ bits in length; verifying integrity of the plaintext blocks using a non-cryptographic Manipulation Detection Function (MDC) function; outputting the plurality of plaintext blocks as an accurate plaintext string if the integrity verification passes; and outputting a failure indicator if the integrity verification fails.

[0107] In a further aspect of the present invention, the program code for causing the performance of the step of selecting $n+1$ ciphertext blocks from the plurality of ciphertext blocks representing n data blocks and one MDC block and performing the reverse hidden-ciphertext randomization function comprises code for: generating a sequence of unpredictable elements for the hidden ciphertext each of ℓ-bit length in the same manner as used at an encryption program product; selecting n+1 ciphertext blocks from the plurality of ciphertext blocks representing n data blocks and one MDC block in the same order as that used at an encryption program product, and combining the selected ciphertext blocks with the sequence of unpredictable elements for the hidden ciphertext to obtain a plurality of hidden ciphertext blocks (z_i), such that each of the n+1 ciphertext blocks identified by index i is combined with the element of the sequence of unpredictable elements for the hidden ciphertext identified by index i, by the inverse of the operation for the hidden ciphertext used at the encryption program product; and wherein the program code for causing the performance of the step of verifying integrity comprises code for creating an MDC decryption block by applying the non-cryptographic Manipulation Detection Code function to the n decrypted plaintext data blocks; and code for comparing the created MDC decryption block with the decrypted MDC block.

[0108] In a further aspect of the present invention, the program code for causing the performance of the step of presenting the hidden ciphertext blocks to a selected parallel decryption mode comprises code for: processing each of the hidden ciphertext blocks with the inverse of the block cipher used at an encryption program product using a secret key (K) to obtain a plurality of hidden plaintext blocks; and performing a reverse plaintext randomization function over the plurality of hidden plaintext blocks to create a plurality of n decrypted plaintext data blocks and one decrypted MDC block each of ℓ bits in length.

[0109] In a further aspect of the present invention, the program code for causing the performance of the reverse plaintext randomization function comprises code for: generating a sequence of unpredictable elements for the hidden plaintext each of ℓ-bit length in the same manner as used at an encryption program product; and combining the selected hidden plaintext blocks with the sequence of unpredictable elements for the hidden plaintext to obtain a plurality of n plaintext blocks and one decrypted MDC block, such that each of the n + 1 hidden plaintext blocks identified by index i is combined with the element of the sequence of unpredictable elements for the hidden plaintext identified by index i, by the inverse of the operation for the hidden plaintext used at the encryption program product.

[0110] In a further embodiment of the present invention, a program product is provided for segmented encryption processing of a message comprising machine-readable program code for causing the performance of the following method steps: partitioning the input plaintext string into a plurality of input plaintext segments; concurrently presenting each different one of the plurality of input plaintext segments to a different one of a plurality of program products for parallel encryption, each of the different program products using a different ℓ-bit secret random number per segment to obtain a ciphertext segment, wherein each encryption program product provides both data confidentiality and integrity with a single processing pass over the input plaintext segment and a single cryptographic primitive, and uses a non-cryptographic Manipulation Detection Code function, wherein the single cryptographic primitive is an ℓ-bit block cipher using a secret key; assembling the plurality of ciphertext segments into a ciphertext string; and outputting the ciphertext string.

[0111] In a further aspect of the present invention, the program code for causing the performance of the step of assembling comprises code for including in the ciphertext string the number of ciphertext segments, a ciphertext segment index, a length of each ciphertext segment and a sequence of ciphertext segments.

[0112] In a further embodiment of the present invention, a program product is provided for segmented decryption processing of a message comprising machine-readable program code for causing a machine to perform the following method steps: presenting a string including the ciphertext string of a message for decryption; partitioning the ciphertext string into a plurality of ciphertext segments; concurrently presenting the plurality of ciphertext segments to a plurality of decryption modes; obtaining a different secret random number per ciphertext segment in the same manner as at the program product for segmented encryption; for decrypting each ciphertext segment using the different secret random number per ciphertext segment to obtain a plaintext segment, using a program product for parallel decryption that is the inverse of a program product for parallel encryption that provides both data confidentiality and integrity with a single processing pass over the input plaintext segment and a single cryptographic primitive, wherein the single cryptographic primitive is an ℓ-bit block cipher using a secret key, and using a non-cryptographic Manipulation Detection Code function for verifying integrity of the plaintext blocks of each plaintext segment; and verifying the integrity of each plaintext segment and for each plaintext segment, outputting either the plaintext segment if the integrity verification passes, or an error indicator.

[0113] In a yet further embodiment of the present invention, a system is disclosed for parallel encryption for providing both data confidentiality and integrity for a message, comprising: a first component for receiving an input plaintext string comprising a message; a second component for generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string; a third component for creating an MDC block of ℓ bits in length that includes the result of applying a non-cryptographic Manipulation Detection Code (MDC) function to the plurality of the equal-size blocks; a fourth component for presenting the equal-size blocks and the MDC block to a selected parallel encryption mode that makes one and only one processing pass with a single cryptographic primitive over each of the equal-size blocks and the MDC block to create a plurality of hidden ciphertext blocks each of ℓ bits in length; and a fifth component for performing a hidden ciphertext randomization function over the plurality of hidden ciphertext blocks to create a plurality of output ciphertext blocks each of ℓ bits in length.

[0114] In a further aspect of the present invention, the fourth component for presenting the equal-size blocks and the MDC block to a selected parallel encryption mode comprises a component for processing each of the equal-size blocks and the MDC block by a parallel encryption mode is confidentiality-secure against chosen-plaintext attacks, wherein each of the equal-size blocks and the MDC block is processed by a block cipher using a secret key (K) to obtain the plurality of hidden ciphertext blocks; and wherein the fifth component for performing a hidden ciphertext randomization function step comprises a component for combining each of the hidden ciphertext blocks with a corresponding element of a sequence of unpredictable elements for the hidden ciphertext to create a set of output blocks of the ciphertext, wherein a hidden ciphertext block identified by an index i is combined with the element of the sequence identified by index i by an operation for the hidden ciphertext that has an inverse.

[0115] In a further aspect of the present invention, the component for processing each of the equal-size blocks and the MDC block by a parallel encryption mode that is confidentiality-secure against chosen-plaintext attacks comprises: a component for performing a plaintext randomization function over the plurality of equal-sized blocks of the plaintext and the MDC block to create a plurality of hidden plaintext blocks each of ℓ bits in length; and a component for processing each of the hidden plaintext blocks by a block cipher using the secret key (K) to obtain the plurality of hidden ciphertext blocks.

[0116] In a further aspect of the present invention, the component for performing a plaintext randomization function step comprises a component for combining each of the equal-size blocks and the MDC block with a corresponding element of a sequence of unpredictable elements for the hidden plaintext to create a set of hidden plaintext blocks, wherein an equal-size block or the MDC block identified by an index i is combined with the element of the sequence identified by index i by an operation for the hidden plaintext that has an inverse.

[0117] In a further aspect of the present invention, the result of the combination of any two different unpredictable elements of the sequence of unpredictable elements for the hidden ciphertext by the inverse operation of the operation for the hidden ciphertext is unpredictable; and wherein the unpredictable elements selected as the two unpredictable elements for the hidden ciphertext are any two different elements of the same sequence of unpredictable elements for the hidden ciphertext used for the encryption of the plaintext string; and wherein the unpredictable elements selected as the two unpredictable elements for the hidden ciphertext are any two different elements of a plurality of sequences of unpredictable elements for the hidden ciphertext used for encryption of a plurality of plaintext strings with the same secret key K.

[0118] In a further aspect of the present invention, the result of the combination of any two different unpredictable elements of the sequence of unpredictable elements for the hidden plaintext by the inverse operation of the operation for the hidden plaintext is unpredictable; and wherein the unpredictable elements selected as the two unpredictable elements for the hidden plaintext are any two different elements of the same sequence of unpredictable elements for the hidden plaintext used for the encryption of the plaintext string; and wherein the unpredictable elements selected as the two unpredictable elements for the hidden plaintext are any two different elements of a plurality of sequences of unpredictable elements for the hidden plaintext used for encryption of a plurality of plaintext strings with the same secret key K.

[0119] In a yet further embodiment of the present invention, a system is disclosed for parallel decryption that is the inverse of a system for parallel encryption which provides both data confidentiality and integrity, comprising: a first component for presenting a string including ciphertext string for decryption; a second component for partitioning the ciphertext string into a plurality of ciphertext blocks comprising ℓ bits each; a third component for selecting $n+1$ ciphertext blocks from the plurality of ciphertext blocks representing n data blocks and one MDC block and performing a reverse hidden ciphertext randomization function on each of the selected $n+1$ ciphertext blocks to obtain a plurality of hidden ciphertext blocks each of ℓ bits in length; a fourth component for presenting the hidden ciphertext blocks to a selected parallel decryption mode that makes one and only one processing pass with a single cryptographic primitive that is the inverse of an encryption single cryptographic primitive over the plurality of hidden ciphertext blocks to obtain a plurality of plaintext blocks and one decrypted MDC block each of ℓ bits in length; a fifth component for verifying integrity of the plaintext blocks using a non-cryptographic Manipulation Detection Function (MDC) function; a sixth component for outputting the plurality of plaintext blocks as an accurate plaintext string if the integrity verification passes; and a seventh component for outputting a failure indicator if the integrity verification fails.

[0120] In a further aspect of the present invention, the third component for selecting $n+1$ ciphertext blocks from the plurality of ciphertext blocks representing n data blocks and one MDC block and performing the reverse hidden-ciphertext randomization function comprises: a component for generating a sequence of unpredictable elements for the hidden ciphertext each of ℓ-bit length in the same manner as used at an encryption system; a component for selecting n + 1 ciphertext blocks from the plurality of ciphertext blocks representing n data blocks and one MDC block in the same order as that used at an encryption system, and combining the selected ciphertext blocks with the sequence of unpredictable elements for the hidden ciphertext to obtain a plurality of hidden ciphertext blocks (z_i), such that each of the n + 1 ciphertext blocks identified by index i is combined with the element of the sequence of unpredictable elements for the hidden ciphertext identified by index i, by the inverse of the operation for the hidden ciphertext used at the encryption system; and wherein the fifth code for verifying integrity step comprises a component for creating an MDC decryption block by applying the non-cryptographic Manipulation Detection Code function to the n decrypted plaintext data blocks; and a component for comparing the created MDC decryption block with the decrypted MDC block.

[0121] In a further aspect of the present invention, the fourth component for presenting the hidden ciphertext blocks to a selected parallel decryption mode comprises: a component for processing each of the hidden ciphertext blocks with the inverse of the block cipher used at an encryption system using a secret key (K) to obtain a plurality of hidden plaintext blocks; and a component for performing a reverse plaintext randomization function over the plurality of hidden plaintext blocks to create a plurality of n decrypted plaintext data blocks and one decrypted MDC block of ℓ bits in length.

[0122] In a further aspect of the present invention, the component for performing the reverse plaintext randomization function comprises: a component for generating a sequence of unpredictable elements for the hidden plaintext each of ℓ-bit length in the same manner as used at an encryption system; and a component for combining the selected hidden plaintext blocks with the sequence of unpredictable elements for the hidden plaintext to obtain a plurality of n plaintext blocks and one decrypted MDC block, such that each of the n+1 hidden plaintext blocks identified by index i is combined with the element of the sequence of unpredictable elements for the hidden plaintext identified by index i, by the inverse of the operation for the hidden plaintext used at the encryption system.

[0123] In a yet further embodiment of the present invention, a system is disclosed for segmented encryption processing of a message comprising: a first component for partitioning the input plaintext string into a plurality of input plaintext segments; a second component for concurrently presenting each different one of the plurality of input plaintext segments to a different one of a plurality of systems for parallel encryption, each of the different systems using a different ℓ-bit secret random number per segment to obtain a ciphertext segment, wherein each encryption system provides both data confidentiality and integrity with a single processing pass over the input plaintext segment and a single cryptographic primitive, and uses a non-cryptographic Manipulation Detection Code function, wherein the single cryptographic primitive is an ℓ-bit block cipher using a secret key; a third component for assembling the plurality of ciphertext segments into a ciphertext string; and a fourth component outputting the ciphertext string.

[0124] In a further aspect of the present invention, the third component for assembling step comprises a component for including in the ciphertext string the number of ciphertext segments, a ciphertext segment index, a length of each ciphertext segment and a sequence of ciphertext segments.

[0125] In a yet further embodiment of the present invention, a system is disclosed for segmented decryption processing of a message comprising: a first component for presenting a string including the ciphertext string of a message for decryption; a second component for partitioning the ciphertext string into a plurality of ciphertext segments; a third component for concurrently presenting the plurality of ciphertext segments to a plurality of decryption modes; a fourth component for obtaining a different secret random number per ciphertext segment in the same manner as at the system for segmented encryption; a fifth component for decrypting each ciphertext segment using the different secret random number per ciphertext segment to obtain a plaintext segment, using a system for parallel decryption that is the inverse of a system for parallel encryption that provides both data confidentiality and integrity with a single processing pass over the input plaintext segment and a single cryptographic primitive, wherein the single cryptographic primitive is an ℓ-bit block cipher using a secret key, and using a non-cryptographic Manipulation Detection Code function for verifying integrity of the plaintext blocks of each plaintext segment; and a sixth component for verifying the integrity of each plaintext segment and for each plaintext segment, outputting either the plaintext segment if the integrity verification passes, or an error indicator.

[0126] In a yet further embodiment of the present invention, a program product is disclosed for a parallel encryption for providing both data confidentiality and integrity for a message, that updates a ciphertext string incrementally, including machine-readable code for performing the following method steps: receiving an input plaintext string comprising a message; generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string; creating an MDC block of ℓ bits in length that includes the result of applying a non-cryptographic Manipulation Detection Code (MDC) function to the plurality of the equal-size blocks; performing a plaintext randomization function over the plurality of equal-sized blocks of the plaintext and the MDC block to create a plurality of hidden plaintext blocks each of ℓ bits in length; processing each of the hidden plaintext blocks by a block cipher using a secret key (K) to obtain a plurality of hidden ciphertext blocks; performing a hidden ciphertext randomization function over the plurality of hidden ciphertext blocks to create a plurality of output ciphertext blocks each of ℓ bits in length; and further including machine-readable code for perfoming the following method steps: receiving an input plaintext string; generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string; receiving an input ciphertext string including a plurality of n+1 equal-size blocks of the ciphertext of ℓ bits in length, wherein the n+1 block of the ciphertext corresponds to an MDC block for the plaintext string; receiving a new ℓ-bit plaintext block to replace an ℓ-bit plaintext block at index i; creating a new MDC block of ℓ bits in length that includes the result of applying a non-cryptographic Manipulation Detection Code (MDC) function to the plurality of the equal-size blocks and the new ℓ-bit plaintext block; performing the same plaintext randomization function as that used at a parallel encryption method over the new ℓ-bit plaintext block and the new MDC block to create two new hidden plaintext blocks each of ℓ bits in length using index i for the new ℓ-bit plaintext block and index n+1 for the new MDC block; processing each of the two new hidden plaintext blocks by a block cipher using the secret key (K) to obtain two new hidden ciphertext blocks; performing the same hidden ciphertext randomization function as that used at a parallel encryption method over the two new hidden ciphertext blocks to create two new output ciphertext blocks each of ℓ bits in length using index i for the new ℓ-bit plaintext block and index n+1 for the new MDC block; replacing in the input ciphertext string, the input ciphertext block at index i with the output ciphertext block for the new ℓ-bit plaintext block and replace the input ciphertext block at index n+1 with the output ciphertext block for the new MDC block, to create a new ciphertext string; and outputting the new ciphertext string.

[0127] In a further aspect of the present invention, the program code for causing the performance of the step of generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string further comprises code for: padding the input plaintext string as necessary such that its length is a multiple of ℓ bits; and partitioning the padded input plaintext string into a plurality of equal-size blocks of ℓ bits in length.

[0128] In a further aspect of the present invention, the program product claim includes machine-readable code for performing the method steps: receiving a plurality of new ℓ-bit plaintext blocks to replace a plurality of ℓ-bit plaintext blocks at the plaintext string at index i; and providing a parallel encryption method that outputs a ciphertext string incrementally for each of the plurality of new ℓ-bit plaintext blocks.

[0129] In a yet further embodiment of the present invention, a program product is disclosed for parallel encryption method for providing both data confidentiality and integrity for a message, including machine-readable program code for causing a machine to perform the method steps: receiving an input plaintext string comprising a message; generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string; partitioning the padded input plaintext string into a plurality of equal-size blocks of ℓ bits in length; creating an MDC block of ℓ bits in length that includes the result of applying a non-cryptographic Manipulation Detection Code (MDC) function to the plurality of the equal-size blocks; performing a plaintext randomization function over the plurality of equal-sized blocks of the plaintext and the MDC block using a different plaintext index for each equal-sized block and the MDC block to create a plurality of hidden plaintext blocks each of ℓ bits in length; processing each of the hidden plaintext blocks by a block cipher using a secret key (K) to obtain a plurality of hidden ciphertext blocks; performing a hidden ciphertext randomization function over the plurality of hidden ciphertext blocks using a different ciphertext index for each hidden ciphertext block to create a plurality of output ciphertext blocks each of ℓ bits in length; and further including machine-readable program code for performing an out-of-order decryption method for the parallel encryption method, which provides both data confidentiality and integrity, including code for: receiving a string including a plurality of $n+1$ ℓ-bit ciphertext blocks for decryption; selecting $n+1$ ciphertext blocks from the plurality of ciphertext blocks representing n data blocks and one MDC block and performing a reverse hidden ciphertext randomization function on each of the selected $n+1$ ciphertext blocks using the ciphertext index to obtain a plurality of hidden ciphertext blocks each of ℓ bits in length; processing each of the hidden ciphertext blocks with the inverse of the block cipher used at an encryption method using the secret key (K) to obtain a plurality of hidden plaintext blocks; and performing an inverse plaintext randomization function over the plurality of hidden plaintext blocks using the plaintext index to create a plurality of n decrypted plaintext data blocks and one decrypted MDC block each of ℓ-bit length; creating an MDC decryption block by applying the non-cryptographic Manipulation Detection Code function to the n decrypted plaintext data blocks in the same manner as at a parallel encryption method; verifying integrity of the plaintext blocks by comparing the created MDC decryption block with the decrypted MDC block; outputting the plurality of plaintext blocks as an accurate plaintext string if the integrity verification passes; and outputting a failure indicator if the integrity verification fails.

[0130] In a further aspect of the present invention, the program code for generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string comprises code for: padding the input plaintext string as necessary such that its length is a multiple of ℓ bits; and partitioning the padded input plaintext string into a plurality of equal-size blocks of ℓ bits in length.

[0131] In a yet further embodiment of the present invention, a system is disclosed for a parallel encryption for providing both data confidentiality and integrity for a message, that updates a ciphertext string incrementally, comprising: a first component for receiving an input plaintext string comprising a message; a second component for generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string; a third component for creating an MDC block of ℓ bits in length that includes the result of applying a non-cryptographic Manipulation Detection Code (MDC) function to the plurality of the equal-size blocks; a fourth component for performing a plaintext randomization function over the plurality of equal-sized blocks of the plaintext and the MDC block to create a plurality of hidden plaintext blocks each of ℓ bits in length; a fifth component for processing each of the hidden plaintext blocks by a block cipher using a secret key (K) to obtain a plurality of hidden ciphertext blocks; a sixth component for performing a hidden ciphertext randomization function over the plurality of hidden ciphertext blocks to create a plurality of output ciphertext blocks each of ℓ bits in length; and further comprising: a seventh component for receiving an input plaintext string; an eight component for generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string; a ninth component for receiving an input ciphertext string including a plurality of $n+1$ equal-size blocks of the ciphertext of ℓ bits in length, wherein the n+1 block of the ciphertext corresponds to an MDC block for the plaintext string; a tenth component for receiving a new ℓ-bit plaintext block to replace an ℓ-bit plaintext block at index i; an eleventh component for creating a new MDC block of ℓ bits in length that includes the result of applying a non-cryptographic Manipulation Detection Code (MDC) function to the plurality of the equal-size blocks and the new ℓ-bit plaintext block; a twelfth component for performing the same plaintext randomization function as that used at a parallel encryption method over the new ℓ-bit plaintext block and the new MDC block to create two new hidden plaintext blocks each of ℓ bits in length using index i for the new ℓ-bit plaintext block and index n+1 for the new MDC block; a thirteenth component for processing each of the two new hidden plaintext blocks by a block cipher using the secret key (K) to obtain two new hidden ciphertext blocks; a fourteenth component for performing the same hidden ciphertext randomization function as that used at a parallel encryption method over the two new hidden ciphertext blocks to create two new output ciphertext blocks each of ℓ bits in length using index i for the new ℓ-bit plaintext block and index n+1 for the new MDC block; a fifteenth component for replacing in the input ciphertext string, the input ciphertext block at index i with the output ciphertext block for the new ℓ-bit plaintext block and replace the input ciphertext block at index n+1 with the output ciphertext block for the new MDC block, to create a new ciphertext string; and a sixteenth component for outputting the new ciphertext string.

[0132] In a further aspect of the present invention, the second component for generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string further comprises: a component for padding the input plaintext string as necessary such that its length is a multiple of ℓ bits; and a component for partitioning the padded input plaintext string into a plurality of equal-size blocks of ℓ bits in length.

[0133] In a further aspect of the present invention, the system further comprises: a component for receiving a plurality of new ℓ-bit plaintext blocks to replace a plurality of ℓ-bit plaintext blocks at the plaintext string at index i; and a component for providing a parallel encryption method that outputs a ciphertext string incrementally for each of the plurality of new ℓ-bit plaintext blocks.

[0134] In a yet further embodiment of the present invention, a system is disclosed for parallel encryption method for providing both data confidentiality and integrity for a message, comprising: a first component for receiving an input plaintext string comprising a message; a second component for generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string; a third component for partitioning the padded input plaintext string into a plurality of equal-size blocks of ℓ bits in length; a fourth component for creating an MDC block of ℓ bits in length that includes the result of applying a non-cryptographic Manipulation Detection Code (MDC) function to the plurality of the equal-size blocks; a fifth component for performing a plaintext randomization function over the plurality of equal-sized blocks of the plaintext and the MDC block using a different plaintext index for each equal-sized block and the MDC block to create a plurality of hidden plaintext blocks each of ℓ bits in length; a sixth component for processing each of the hidden plaintext blocks by a block cipher using a secret key (K) to obtain a plurality of hidden ciphertext blocks; a seventh component for performing a hidden ciphertext randomization function over the plurality of hidden ciphertext blocks using a different ciphertext index for each hidden ciphertext block to create a plurality of output ciphertext blocks each of ℓ bits in length; and further comprising for performing an out-of-order decryption method for the parallel encryption method, which provides both data confidentiality and integrity: an eighth component for receiving a string including a plurality of $n+1$ ℓ-bit ciphertext blocks for decryption; a ninth component for selecting $n+1$ ciphertext blocks from the plurality of ciphertext blocks representing n data blocks and one MDC block and performing a reverse hidden ciphertext randomization function on each of the selected $n+1$ ciphertext blocks using the ciphertext index to obtain a plurality of hidden ciphertext blocks each of ℓ bits in length; a tenth component for processing each of the hidden ciphertext blocks with the inverse of the block cipher used at an encryption method using the secret key (K) to obtain a plurality of hidden plaintext blocks; and an eleventh component for performing an inverse plaintext randomization function over the plurality of hidden plaintext blocks using the plaintext index to create a plurality of n decrypted plaintext data blocks and one decrypted MDC block each of ℓ-bit length; a twelfth component for creating an MDC decryption block by applying the non-cryptographic Manipulation Detection Code function to the n decrypted plaintext data blocks in the same manner as at a parallel encryption method; a thirteenth component for verifying integrity of the plaintext blocks by comparing the created MDC decryption block with the decrypted MDC block; a fourteenth component for outputting the plurality of plaintext blocks as an accurate plaintext string if the integrity verification passes; and a fifteenth component for outputting a failure indicator if the integrity verification fails.

[0135] In a further aspect of the present invention, the second component for generating a plurality of equal-sized blocks of ℓ bits in length from the input plaintext string comprises: a component for padding the input plaintext string as necessary such that its length is a multiple of

ℓ bits; and a component for partitioning the padded input plaintext string into a plurality of equal-size blocks of ℓ bits in length.

BRIEF DESCRIPTION OF THE DRAWINGS

[0136] For a more complete understanding of the present invention and the advantages thereof, reference should be made to the following Detailed Description taken in connection with the accompanying drawings, in which:

[0137] Figure 1 illustrates a schematic diagram of the method of the present invention for the parallel encryption of input plaintext string $x = x_1\ x_2\ x_3\ x_4$, using secret key K to obtain output ciphertext string $y = y_0\ y_1\ y_2\ y_3\ y_4\ y_5$.

[0138] Figure 2 illustrates a schematic diagram of the method of the present invention for the parallel decryption of the input ciphertext string $y = y_0\ y_1\ y_2\ y_3\ y_4\ y_5$, using secret key K to obtain the output plaintext string $x = x_1\ x_2\ x_3\ x_4$ or the error indicator.

[0139] Figure 3 illustrates a schematic diagram for the preferred embodiment of this invention of the stateless parallel encryption mode in which input plaintext string $x = x_1\ x_2\ x_3\ x_4$ is encrypted using secret key K to obtain output ciphertext $y = y_0\ y_1\ y_2\ y_3\ y_4\ y_5$.

[0140] Figure 4 illustrates a schematic diagram for the preferred embodiment of this invention of the stateless parallel decryption mode in which input ciphertext string $y = y_0\ y_1\ y_2\ y_3\ y_4\ y_5$ is decrypted using secret key K to obtain output plaintext string $x = x_1\ x_2\ x_3\ x_4$ or the error indicator.

[0141] Figure 5 illustrates a schematic diagram for the preferred embodiment of this invention of the stateful-sender parallel encryption mode in which input plaintext string $x = x_1\ x_2\ x_3\ x_4$ is encrypted using secret key K to obtain output ciphertext $y = y_1\ y_2\ y_3\ y_4\ y_5$.

[0142] Figure 6 illustrates a schematic diagram for the preferred embodiment of this invention of the stateful-sender parallel decryption mode in which input ciphertext string $y = y_1\ y_2\ y_3\ y_4\ y_5$ is decrypted using secret key K to obtain output plaintext string $x = x_1\ x_2\ x_3\ x_4$ or the error indicator.

[0143] Figure 7 illustrates a schematic diagram for the preferred embodiment of this invention of the stateful parallel encryption mode in which input plaintext string $x = x_1\ x_2\ x_3\ x_4$ is encrypted using secret key K to obtain output ciphertext $y = y_1\ y_2\ y_3\ y_4\ y_5$.

[0144] Figure 8 illustrates a schematic diagram for the preferred embodiment of this invention of the stateful parallel decryption mode in which input ciphertext string $y = y_1\ y_2\ y_3\ y_4\ y_5$ is decrypted using secret key K to obtain output plaintext string $x = x_1\ x_2\ x_3\ x_4$ or the error indicator.

[0145] Figure 9 illustrates a schematic diagram for the preferred embodiment of the three-segment stateful-sender parallel encryption mode in which input plaintext string $x = x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\ x_8\ x_9\ x_{10}\ x_{11}\ x_{12}$ is encrypted using secret key K to obtain output ciphertext string $y = y_1\ y_2\ y_3\ y_4\ y'_5\ y_5\ y_6\ y_7\ y_8\ y'_9\ y_9\ y_{10}\ y_{11}\ y_{12}\ y'_{13}$.

[0146] Figure 10 illustrates a schematic diagram for the preferred embodiment of the three-segment stateful-sender parallel decryption mode in which input ciphertext string $y = y_1\ y_2\ y_3\ y_4\ y'_5\ y_5\ y_6\ y_7\ y_8\ y'_9\ y_9\ y_{10}\ y_{11}\ y_{12}\ y'_{13}$ is decrypted using secret key K to obtain a plurality of output plaintext segment $x_1\ x_2\ x_3\ x_4$ or a first error indicator, output plaintext segment $x_5\ x_6\ x_7\ x_8$ or a second error indicator, and output plaintext segment $x_9\ x_{10}\ x_{11}\ x_{12}$ or a third error indicator.

[0147] Figure 11 illustrates a schematic diagram for the preferred embodiment of the three-segment stateful parallel encryption mode in which input plaintext string $x = x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\ x_8\ x_9\ x_{10}\ x_{11}\ x_{12}$ is encrypted using secret key K to obtain output ciphertext string y = y_1 y_2 y_3 y_4 y'_5 y_5 y_6 y_7 y_8 y'_9 y_9 y_{10} y_{11} y_{12} y'_{13}.

[0148] Figure 12 illustrates a schematic diagram for the preferred embodiment of the three-segment stateful parallel decryption mode in which input ciphertext string y = y_1 y_2 y_3 y_4 y'_5 y_5 y_6 y_7 y_8 y'_9 y_9 y_{10} y_{11} y_{12} y'_{13} is decrypted using secret key K to obtain a plurality of output plaintext segment x_1 x_2 x_3 x_4 or a first error indicator, output plaintext segment x_5 x_6 x_7 x_8 or a second error indicator, and output plaintext segment x_9 x_{10} x_{11} x_{12} or a third error indicator.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0149] Referring to Figure 1, a plaintext string x 23 representing the input data is presented to the parallel encryption mode system providing data confidentiality and integrity 51 resulting in an output ciphertext string y 26. It is assumed that the sender and the receiver share a secret key K (31) and that a random-number generator 70 is available. From the input plaintext string x 23, a plurality of equal-sized blocks 21 of ℓ bits in length is generated. In one embodiment, the input plaintext string x 23 is padded so that it is a multiple of ℓ bits. It is assumed that the plaintext string x 23 is composed of n ℓ-bit plaintext blocks 21. Figure 1 shows an example plaintext string 23 composed of n = 4 blocks, x = x_1 x_2 x_3 x_4.

[0150] F is an ℓ-bit block cipher with key length k, F_K 41 is the ℓ-bit block cipher F using secret key K 31. $F_K(b)$ is an ℓ-bit block representing the enciphering of the ℓ-bit block b by F_K.

[0151] The random-number generator 70 outputs a secret random number r_0 71 of ℓ bits in length that is further enciphered by F_K 41, the block cipher F using the first key K 31, to obtain the block y_0 25. In an alternate embodiment, the secret random number r_0 71 is shared between the sender and the receiver, and hence it need not be generated by a random-number generator 70. In the alternate embodiment the sender and the receiver generate the same shared secret random number r_0 71 from an already shared secret key K 31 using key separation techniques well-known in the art.

[0152] The input plaintext blocks 21 are combined using a non-cryptographic Manipulation Detection Code (MDC) function 91 yielding an ℓ-bit MDC block. Examples of the result MDC(x) are provided below. By way of example, the non-cryptographic MDC function is a high-performance MDC function. In the preferred embodiment of this invention, this function is a bit-wise exclusive-or function. In the example of Figure 1, in which the input plaintext string 23 is $x = x_1\ x_2\ x_3\ x_4$, $MDC(x) = x_1 \oplus x_2 \oplus x_3 \oplus x_4$, where $\oplus$ denotes the bit-wise exclusive-or operation. In an alternate embodiment of this invention, the non-cryptographic MDC function uses addition modulo $2^{\ell} - 1$; i.e., for the example of Figure 1 in which the input plaintext string is $x = x_1\ x_2\ x_3\ x_4$, $MDC(x) = x_1 + x_2 + x_3 + x_4$ (modulo 2^{ℓ} - 1). In yet another alternate embodiment of this invention, the non-cryptographic MDC function is any other parity checking code such as a cyclic redundancy code function. In the preferred embodiment of this invention, the result of the application of the MDC function, MDC(x), represents the ℓ-bit MDC block 22. In an alternate embodiment, the result of the application of the MDC function, MDC(x), is further combined with a secret random vector z_0 that is obtained by enciphering with F_K, the block cipher F using the first key K, of a variant, $r_0 + c$, of the random number r_0 71, where c is a non-zero constant, the combination resulting in the block value $MDC(x) \oplus z_0$, which represents the computed ℓ-bit MDC block 22. In this alternate embodiment of this invention, the combination operation between MDC(x) and the secret random vector z_0 is the bit-wise exclusive-or operation denoted by $\oplus$; i.e.

the resulting value 22 is MDC(x) $\oplus$ z_0. In another alternate embodiment of this invention, the combination operation between MDC(x) and the secret random vector z_0 is the addition modulo $2^{\ell} - 1$; i.e., the resulting value 22 is MDC(x) + z_0 (modulo 2^{ℓ} - 1).

[0153] The plurality of input plaintext blocks 21 and the MDC block 22 are submitted to a selected parallel encryption mode 61 that uses a block cipher F_K with key K 31. In an aspect of this invention, the selected parallel encryption mode 61 is confidentiality-secure. In a further aspect of this invention, the selected confidentiality-secure parallel encryption mode 61 has the property that the input plaintext blocks 21 and the block value MDC(x) 22 are part of the input to F_K, the block cipher F using the first key K 31, used by the selected confidentiality-secure encryption mode 61.

[0154] The application of the selected parallel encryption mode 61 results in a plurality of hidden ciphertext blocks 87 of ℓ-bit length; the number of hidden ciphertext blocks 87 is greater by one than the number of the input plaintext blocks 21; i.e., it is n+1. For the example of Figure 1, wherein n = 4, the plurality of hidden ciphertext blocks 87 comprises n+1=5 blocks z_1, z_2, z_3, z_4, z_5. These hidden ciphertext blocks 87 are submitted to a hidden ciphertext randomization step comprising, in one embodiment, applying a combination operation for the hidden ciphertext 84 to each hidden ciphertext block z_i 87 and each ℓ-bit element E_i 83 of a sequence of n+1 elements for the hidden ciphertext.

[0155] Each of the elements E_i 83 is unpredictable because it is obtained by combining the secret random number r_0 71 and the element identifier i such that for any given ℓ-bit constant a, the probability of the event E_i = a is negligible, wherein the notion of negligible probability is well-known to those skilled in the art (viz., M. Naor and O. Reingold:

``From Unpredictability to Indistinguishability: A Simple Construction of Pseudo-Random Functions from MACs,'' Advances in Cryptology - CRYPTO '98 (LNCS 1462), pp. 267-282, 1998; M. Bellare, A. Desai, E. Jokipii, and P. Rogaway: ``A Concrete Security Treatment of Symmetric Encryption,'' Proceedings of the 38th Symposium on Foundations of Computer Science, IEEE, 1997, pp. 394-403). The fact that these elements E_i 83 are unpredictable means that enough of their ℓ bits remain unknown so that the probability of the event $E_i = a$ is negligible. In the preferred embodiment of this invention, each unpredictable element E_i 83 is obtained by multiplication modulo 2^ℓ of the element index i and the secret random number r_0 71; i.e., $E_i = r_0 \times i$. In an alternate embodiment, when encryption is performed sequentially, each element of the sequence E_{i+1} (where $i \geq 1$) is generated from the previous element E_i by modular 2^ℓ addition of the secret random number r_0, the first element of the sequence being the secret random number r_0 itself, namely $E_1 = r_0$. It should be appreciated by those skilled in the art, and is a further aspect of this invention, that the unpredictable elements 83 and the combination operation 84 can be obtained in other ways that do not depart from the spirit and scope of the present invention as set forth in the claims. In an alternate embodiment of this invention, the unpredictable elements E_i are the elements of the linear congruence sequence defined by $E_i = a^i \times r_0$, where a is called the multiplier and is chosen to pass all the necessary spectral tests, i is the element index, $i = 1, \ldots, n+1$, and r_0 is the secret random number 71, as described by D.E. Knuth in ``The Art of Computer Programming - Volume 2: Seminumerical Algorithms,'' Addison-Wesley, 1981 (second edition), Chapter 3, incorporated herein by reference.

[0156] The combination operation for the hidden ciphertext 84 is an operation that has an inverse. In the preferred embodiment of this invention, the combination operation 84 is the modular 2^{ℓ} addition, whereby each ciphertext block is obtained as $y_i = z_i + E_i$ modulo 2^{ℓ}. In an alternate embodiment of this invention, the combination operation 84 is the bit-wise exclusive-or operation, whereby each ciphertext block $y_i = z_i \oplus E_i$. In yet another alternate embodiment of this invention, the combination operation 84 is modular 2^{ℓ} subtraction operation, whereby each ciphertext block $y_i = z_i - E_i$ modulo 2^{ℓ}. The invention, however, is not so limited, as other combination operations that have an inverse may also be used for combination operation for the hidden ciphertext 84.

[0157] In the preferred embodiment of this invention, the distinct unpredictable elements E_i 83 (where $i \geq 1$) and the combination operation for the hidden ciphertext 84 are chosen such that for any two distinct unpredictable elements E_i, E_j, both used for the same message or each used for different messages encrypted with the same key K 31, the combination E_i op^{-1} E_j results in an ℓ-bit block that is unpredictable, where op^{-1} denotes the inverse of the combination operation 84. That is, for any given ℓ-bit constant a, the probability of the event E_i op^{-1} E_j = a is negligible, wherein the notion of negligible probability is well-known to those skilled in the art (viz., M. Naor and O. Reingold: ``From Unpredictability to Indistinguishability: A Simple Construction of Pseudo-Random Functions from MACs,'' Advances in Cryptology - CRYPTO '98 (LNCS 1462), pp. 267-282, 1998; M. Bellare, A. Desai, E. Jokipii, and P. Rogaway: ``A Concrete Security Treatment of Symmetric Encryption,'' Proceedings of the 38th Symposium on Foundations of Computer Science, IEEE, 1997, pp. 394-403). The fact that block E_i op^{-1} E_j is unpredictable means that enough of its ℓ bits remain unknown so that the probability of the event E_i op^{-1} Ej = a is negligible.

[0158] The application of the combination operation 84 to the plurality of hidden ciphertext blocks 87 and the unpredictable elements 83 of the sequence results in a plurality of ciphertext blocks y_i 24. Ciphertext block y_0 25 and the plurality of ciphertext blocks y_i 24 form the ciphertext string y 26 that has n + 2 blocks and is the output data of the encryption mode 51. For the example presented in Figure 1, the ciphertext string 26 is $y = y_0\ y_1\ y_2\ y_3\ y_4\ y_5$; i.e., has $n+2=6$ blocks.

[0159] Figure 2 represents the decryption of a ciphertext string y 26 composed of block y_0 25 and n + 1 ciphertext blocks 24 to either a plaintext string x 23 composed of n plaintext blocks 21 or an error indicator 20 by the parallel decryption mode providing data confidentiality and integrity 52. Figure 2 shows an example wherein the ciphertext string y 26 is composed of block y_0 25 and $n+1=5$ ciphertext blocks 24; i.e., $y = y_0\ y_1\ y_2\ y_3\ y_4\ y_5$, and the plaintext string x 23 has $n=4$ blocks; i.e., $x = x_1\ x_2\ x_3\ x_4$. It is assumed that the sender shares the secret key K (31) with the receiver of the data string y 26.

[0160] F^{-1}_K 42 is the inverse of the ℓ-bit block cipher F using secret key K 31. F^{-1}_K (d) is an ℓ-bit block representing the deciphering of the ℓ-bit block d by F^{-1}_K.

[0161] Block y_0 25 is deciphered using F^{-1}_K 42, the inverse of the block cipher F using secret key K 31, resulting in the secret random number r_0 71.

[0162] The n + 1 ciphertext blocks y_i 24, where $i \geq 1$, are submitted to the inverse combination operation for the hidden ciphertext 85 together with the unpredictable elements E_i 83, computed at decryption, resulting in n + 1 hidden ciphertext blocks z_i 87. The unpredictable elements E_i 83 are computed exactly in the same way as at parallel encryption (viz., Figure 1). The inverse combination operation for the hidden ciphertext 85 is the inverse of the combination operation for the hidden ciphertext 84 used at encryption. In the preferred embodiment of this invention, if the combination operation 84 is a modular 2^{ℓ} addition operation, then the inverse combination operation 85 is the modular 2^{ℓ} subtraction; i.e., each block $z_i = y_i - E_i$ modulo 2^{ℓ}. In an alternate embodiment of this invention, if the combination operation 84 is the bit-wise exclusive-or operation, then the inverse combination operation 85 is the bit-wise exclusive-or operation; i.e., each block $z_i = y_i \oplus E_i$. In yet another alternate embodiment of this invention, if the combination operation 84 is modular 2^{ℓ} subtraction operation, then the inverse combination operation 85 is the modular 2^{ℓ} addition; i.e., each block $z_i = y_i + E_i$ rnodulo 2^{ℓ}.

[0163] The n+1 hidden ciphertext blocks z_i 87 are sent to the parallel decryption function of the selected mode 62 that uses F^{-1}_K, the inverse of the block cipher F using key K 31. The decryption of the selected mode 61 outputs n plaintext blocks and one decrypted MDC block 29. For the example presented in Figure 2, the n=4 plaintext blocks are x_1, x_2, x_3, x_4 and the decrypted MDC block 29 is x_5. Further, the non-cryptographic MDC function is applied to the n plaintext blocks and the result is MDC(x). In the preferred embodiment of this invention, MDC(x) is the computed MDC block 91. In an alternate embodiment, the result MDC(x) is further combined with the secret vector z_0 to yield the computed ℓ-bit MDC block, $MDC(x) \oplus z_0$ 91, wherein the secret random vector z_0 is obtained from the secret number r_0 by enciphering the variant $r_0 + c$ using F_K, where c is a non-zero constant. Then the computed MDC block 91 and the decrypted MDC block 29 are compared for equality using the comparator 92. If the computed MDC block 91 and the decrypted MDC block 29 are not equal, then the result of the decryption of the data string y 26 is the error indicator 20. If the computed MDC block 91 and the decrypted MDC block 29 are equal, then the output from the logical "and" operators 93 is the result of the decryption of the ciphertext string y 26 using the parallel decryption mode 52; i.e., the result is the plaintext string x 23 comprising n plaintext blocks x_i 21. In the example presented in Figure 2, if computed MDC block 91 and the decrypted MDC block 29 are equal, then the output of the parallel decryption mode 52 is the plaintext string 23 $x = x_1 x_2 x_3 x_4$.

[0164] Figure 3 illustrates a schematic diagram for the preferred embodiment of this invention of the stateless parallel encryption mode. The input plaintext string x 23 (which is padded in a standard way as necessary) containing n plaintext blocks x_i 21 is encrypted using the encryption mode 51 and the result of this encryption is the ciphertext string y 26 containing n+2 ciphertext blocks, namely ciphertext block y_0 25 and n+1 ciphertext blocks y_i 24 where $i = 1, 2, \ldots, n + 1$. The encryption uses a secret key K (31). The random-number generator 70 outputs the secret random number r_0 71 that is further enciphered with F_K 41, the block cipher F using the first key K 31, and the result is ciphertext block y_0 25.

[0165] In this embodiment, the plaintext blocks x_i 21 are bit-wise exclusive-or-ed into MDC(x) 22; i.e., $MDC(x) = x_1 \oplus \ldots \oplus x_n$, and this value is appended to the plaintext string x and submitted to selected parallel encryption mode 61 that uses F_K, the block cipher F using the key K 31. The parallel encryption mode 61 comprises a plaintext randomization step applied to the n plaintext blocks x_i 21 and the MDC block 22 to generate the hidden plaintext blocks v_i 88 that are further enciphered with F_K, the block cipher F using the first key K 31, resulting in n + 1 hidden ciphertext blocks z_i 87. Figure 3 shows an example where n = 4; i.e. the hidden plaintext blocks v_i 88 are v_1, v_2, v_3, v_4, v_5 and the hidden ciphertext blocks 87 are z_1, z_2, z_3, z_4, z_5.

[0166] In the preferred embodiment of this invention of the stateless encryption, the plaintext randomization step comprises combining each of the plaintext blocks x_i 21 and the MDC block 22, and each ℓ-bit element E_1, E_2, ..., E_n and E^*_{n+1} 81 of a sequence of n+1 elements for the hidden plaintext using a combination operation for the hidden plaintext 82. Each of these elements E_1, E_2, ..., E_n and E^*_{n+1} 81 for the hidden plaintext is unpredictable because it is obtained by combining the secret random number r_0 71 and the element identifier i such that for any given ℓ-bit constant a, the probability of the event equating the i-th element and constant a is negligible, wherein the notion of negligible probability is well-known to those skilled in the art (viz., M. Naor and O. Reingold: ``From Unpredictability to Indistinguishability: A Simple Construction of Pseudo-Random Functions from MACs,'' Advances in Cryptology - CRYPTO '98 (LNCS 1462), pp. 267-282, 1998; M. Bellare, A. Desai, E. Jokipii, and P. Rogaway: ``A Concrete Security Treatment of Symmetric Encryption,'' Proceedings of the 38th Symposium on Foundations of Computer Science, IEEE, 1997, pp. 394-403). In the preferred embodiment of this invention, each unpredictable element 81 is obtained by multiplication modulo 2^{ℓ} of the element index i with the secret random number r_0 71 for each plaintext block and by multiplication modulo 2^{ℓ} of the constant n + 2 with the secret random number r_0 71 for the MDC block, i.e., $E_i = r_0 \times i$ for plaintext blocks with i = 1, 2, ..., n, and $E^*_{n+1} = r_0 \times (n+2)$ for the MDC block. It should be appreciated by those skilled in the art, and is a further aspect of this invention, that the unpredictable elements 81 for the hidden plaintext and the combination operation 82 can be obtained in other ways that do not depart from the spirit and scope of the present invention as set forth in the claims. In an alternate embodiment of this invention, the unpredictable elements 81 for the hidden ciphertext are the elements of the linear congruence sequence defined by $E_i = a^i \times r_0$, for the n plaintext blocks and $E^*_{n+1} = a^{n+2} \times r_0$, where a is called the multiplier and is chosen to pass all the necessary spectral tests, i is the element index, i = 1, ..., n, and r_0 is the secret random number 71, as described by D.E. Knuth in ``The Art of Computer Programming - Volume 2: Seminumerical Algorithms,'' Addison-Wesley, 1981 (second edition), Chapter 3, incorporated herein by reference.

[0167] The combination operation for the hidden plaintext 82 is an operation that has an inverse. In the preferred embodiment of this invention, the combination operation 82 is the modular 2^{ℓ} addition, whereby each hidden plaintext block is obtained as $v_i = x_i + E_i$ modulo 2^{ℓ} for i = 1, 2, ..., n, and $v_{n+1} = x_{n+1} + E^*_{n+1}$ modulo 2^{ℓ} for the MDC block. In an alternate embodiment of this invention, the combination operation 82 is the bit-wise exclusive-or operation. In yet another alternate embodiment of this invention, the combination operation 82 is the modular 2^{ℓ} subtraction operation. The invention, however, is not so limited, as other combination operations that have an inverse may also be used for operation for the hidden plaintext 82.

[0168] In the preferred embodiment of this invention, the distinct unpredictable elements $E_1, E_2, \ldots, E_n$ and E^*_{n+1} 81 (where $i \geq 1$) and the combination operation for the hidden ciphertext 82 are chosen such that for any two distinct unpredictable elements 81, both used for the same message or each used for different messages encrypted with the same key K 31, the combinations E_i op^{-1} E_j and E_i op^{-1} E^*_{n+1} result in ℓ-bit blocks that are unpredictable, where op^{-1} denotes the inverse of the combination operation. That is, for any given ℓ-bit constant a, the probability of event $E_i\ op^{-1}\ E_j = a$ and event $E_i\ op^{-1}\ E^*_{n+1} = a$ is negligible, wherein the notion of negligible probability is well-known to those skilled in the art (viz., M. Naor and O. Reingold: ``From Unpredictability to Indistinguishability: A Simple Construction of Pseudo-Random Functions from MACs,'' Advances in Cryptology - CRYPTO '98 (LNCS 1462), pp. 267-282, 1998; M. Bellare, A. Desai, E. Jokipii, and P. Rogaway: ``A Concrete Security Treatment of Symmetric Encryption,'' Proceedings of the 38th Symposium on Foundations of Computer Science, IEEE, 1997, pp. 394-403). The fact that blocks $E_i\ op^{-1}\ E_j$ and $E_i\ op^{-1}\ E^*_{n+1}$ are unpredictable means that enough of their ℓ bits remain unknown so that the probability of the event $E_i\ op^{-1}\ Ej = a$ and event $E_i\ op^{-1}\ E^*_{n+1} = a$ is negligible.

[0169] In the preferred embodiment of this invention of the stateless parallel encryption, the hidden ciphertext blocks z_i 87 are submitted to a randomization step for the hidden ciphertext comprising applying a combination operation 84 for the hidden ciphertext to each hidden ciphertext block z_i 87 and each ℓ-bit element E_i 83 of a sequence of n+1 elements. Each of these elements E_i 83 is unpredictable because it is obtained by combining the secret random number r_0 71 and the element identifier i such that for any given ℓ-bit constant a, the probability of the event $E_i = a$ is negligible, wherein the notion of negligible probability is well-known to those skilled in the art (viz., M. Naor and O. Reingold: ``From Unpredictability to Indistinguishability: A Simple Construction of Pseudo-Random Functions from MACs,'' Advances in Cryptology - CRYPTO '98 (LNCS 1462), pp. 267-282, 1998; M. Bellare, A. Desai, E. Jokipii, and P. Rogaway: ``A Concrete Security Treatment of Symmetric Encryption,'' Proceedings of the 38th Symposium on Foundations of Computer Science, IEEE, 1997, pp. 394-403). In the preferred embodiment of this invention, each unpredictable element for the hidden ciphertext 83 is obtained by multiplication modulo 2^{ℓ} of the element index i with the secret random number r_0 71; i.e., $E_i = r_0 \times i$ for $i = 1, 2, \ldots, n + 1$. It should be appreciated by those skilled in the art, and is a further aspect of this invention, that the unpredictable elements for the hidden ciphertext 83 and the combination operation for the hidden ciphertext 84 can be obtained in other ways that do not depart from the spirit and scope of the present invention as set forth in the claims. In an alternate embodiment of this invention, the unpredictable elements 83 are the elements of the linear congruence sequence defined by $a^{i} \times r_0$, where a is called the multiplier and is chosen to pass all the necessary spectral tests, i is the element index, $i = 1, \ldots, n+1$, and r_0 is the secret random number 71, as described by D.E. Knuth in ``The Art of Computer Programming - Volume 2: Seminumerical Algorithms,'' Addison-Wesley, 1981 (second edition), Chapter 3, incorporated herein by reference.

[0170] The combination operation for the hidden ciphertext 84 is an operation that has an inverse. In the preferred embodiment of this invention, the combination operation 84 is the modular 2^{ℓ} addition, whereby each ciphertext block is obtained as $y_i = z_i + E_i$ modulo 2^{ℓ}. In an alternate embodiment of this invention, the combination operation 84 is the bit-wise exclusive-or operation. In yet another alternate embodiment of this invention, the combination operation 84 is the modular 2^{ℓ} subtraction operation. The invention, however, is not so limited, as other combination operations that have an inverse may also be used for operation for the hidden ciphertext 84.

[0171] In the preferred embodiment of this invention, the distinct unpredictable elements E_i 83 (where $i \geq 1$) and the combination operation for the hidden ciphertext 84 are chosen such that for any two distinct unpredictable elements E_i, E_j, both used for the same message or each used for different messages encrypted with the same key K 31, the combination E_i op^{-1} E_j results in an ℓ-bit block that is unpredictable, where op^{-1} denotes the inverse of the combination operation. That is, for any given ℓ-bit constant a, the probability of the event E_i op^{-1} E_j = a is negligible, wherein the notion of negligible probability is well-known to those skilled in the art (viz., M. Naor and O. Reingold: ``From Unpredictability to Indistinguishability: A Simple Construction of Pseudo-Random Functions from MACs,'' Advances in Cryptology - CRYPTO '98 (LNCS 1462), pp. 267-282, 1998; M. Bellare, A. Desai, E. Jokipii, and P. Rogaway: ``A Concrete Security Treatment of Symmetric Encryption,'' Proceedings of the 38th Symposium on Foundations of Computer Science, IEEE, 1997, pp. 394-403). The fact that block E_i op^{-1} E_j is unpredictable means that enough of its ℓ bits remain unknown so that the probability of the event E_i op^{-1} Ej = a is negligible.

[0172] The application of the combination operation for the hidden ciphertext 84 to the plurality of hidden ciphertext blocks 87 and the unpredictable elements for the hidden ciphertext, E_i 83, of the sequence results in a plurality of ciphertext blocks y_i 24. Ciphertext block y_0 25 and the plurality of ciphertext blocks y_i 24 form the ciphertext string y 26 that has n+2 blocks and is the output data of the encryption mode 51. For the example presented in Figure 3, the ciphertext string 26 is y = y_0 y_1 y_2 y_3 y_4 y_5; i.e., has n+2=6 blocks.

[0173] Figure 4 illustrates a schematic diagram for the preferred embodiment of this invention of the stateless parallel decryption. From the ciphertext string y 26, ciphertext block y_0 25 is deciphered using the inverse of the block cipher with key K 31, namely F^{-1}_K 42 to obtain the secret random vector r_0 71.

[0174] The secret random number r_0 71 is used to obtain the unpredictable elements for the hidden ciphertext $E_i = r_0 \times i$ (modulo 2^ℓ) 83 in the same way as at encryption (viz., Figure 3). These unpredictable elements E_i 83 and the ciphertext blocks y_i 24 are combined using the inverse combination operation for the ciphertext 85 to generate the hidden ciphertext blocks z_i 87. The inverse combination operation for the hidden ciphertext 85 is the inverse of the combination operation for the hidden ciphertext 84 used at encryption. In the preferred embodiment of this invention of the stateless parallel decryption, the inverse combination operation for the ciphertext 85 is subtraction modulo 2^ℓ; i.e., $z_i = y_i - E_i$. In an alternate embodiment of this invention, when the combination operation 84 is the bit-wise exclusive-or operation, the inverse combination operation for the hidden ciphertext 85 is the bit-wise exclusive-or operation; i.e., $z_i = y_i \oplus E_i$. In another alternate embodiment of this invention, when the combination operation 84 is the modular 2^ℓ subtraction operation, the inverse combination operation for the ciphertext 85 is addition modulo 2^ℓ; i.e., $z_i = y_i + E_i$. The invention, however, is not so limited, as other inverse combination operations may also be used for operation 85, the only restriction being that operation 85 is the inverse of the combination operation for the hidden ciphertext 84.

[0175] The n+1 hidden ciphertext blocks z_i 87 are presented to the select parallel decryption mode 62 that uses F^{-1}_K, the inverse of the block cipher F using key K 31. The parallel decryption mode 62 consists of deciphering the n+1 hidden ciphertext blocks z_i 87 using F^{-1}_K, the inverse of the block cipher F using key K 31 to obtain n+1 hidden plaintext blocks v_i 88 that are further submitted to a reverse plaintext randomization step that generates n+1 blocks x_i. The last block x_{n+1} 29 represents the decrypted MDC block.

[0176] The reverse plaintext randomization step consists of applying the inverse operation for the hidden plaintext 86 to the n+1 hidden plaintext blocks v_i 88 and the n+1 unpredictable elements for the hidden plaintext $E_1, E_2, \ldots, E_n$ and E^*_{n+1} 81 obtained in the same way as at encryption (viz., Figure 3). The inverse combination operation for the hidden plaintext 86 is the inverse of the combination operation for the hidden plaintext 82 used at encryption. In the preferred embodiment of this invention of the stateless parallel decryption, the inverse combination operation for the plaintext 86 is subtraction modulo 2^{ℓ}; i.e., $x_i = v_i - E_i$, for $1 \leq i \leq n$, and $x_{n+1} = v_{n+1} - E^*_{n+1}$ for $i = n + 1$. In an alternate embodiment of this invention, when the combination operation 82 is the bit-wise exclusive-or operation, the inverse combination operation for the hidden plaintext 85 is the bit-wise exclusive-or operation; i.e., $x_i = v_i \oplus E_i$, for $1 \leq i \leq n$, and $x_{n+1} = v_{n+1} \oplus E^*_{n+1}$ for $i = n + 1$. In another alternate embodiment of this invention, when the combination operation 82 is the modular 2^{ℓ} subtraction operation, the inverse combination operation for the hidden plaintext 86 is addition modulo 2^{ℓ}; i.e., $x_i = v_i + E_i$, for $1 \leq i \leq n$, and $x_{n+1} = v_{n+1} + E^*_{n+1}$ for $i = n + 1$. The invention, however, is not so limited, as other inverse combination operations may also be used for operation 86, the only restriction being that operation 86 is the inverse of the combination operation for the hidden plaintext 82.

[0177] The n blocks x_i, namely $x_1, x_2, \ldots, x_n$, in accordance with one embodiment of the MDC function, are bit-wise exclusive-or-ed to obtain computed MDC(x) block 91; i.e. $\mathrm{MDC}(x) = x_1 \oplus \ldots \oplus x_n$. Then the computed MDC(x) an the decrypted MDC block x_{n+1} 29 are compared for equality at 92. If the computed MDC block MDC(x) 91 and the decrypted MDC block 29 are not equal then the result of the decryption of the data string y 26 is the error indicator 20. If the computed MDC block MDC(x) 91 and the decrypted MDC block 29 are equal then the output from the logical "and" operators 93 is the result of the decryption of the ciphertext string y 26 using the decryption mode 52; i.e., the result is the plaintext string x 23 composed of n plaintext blocks x_i 21. For the example illustrated in Figure 4, the output of the parallel decryption mode 52 is the plaintext string 23 $x = x_1 x_2 x_3 x_4$.

[0178] Figure 5 illustrates a schematic diagram for the preferred embodiment of this invention of the stateful-sender parallel encryption mode. The encryption mode 53 uses a secret key K (31). In this embodiment of the method of the invention a counter initialized to a constant, ctr 72, is enciphered using F_K 41, the block cipher F using the first key K 31, to yield the secret random number r_0 71.

[0179] In this embodiment, the plaintext blocks x_i 21 are bit-wise exclusive-or-ed into MDC(x) 22; i.e., $MDC(x) = x_1 \oplus \ldots \oplus x_n$, and this value is appended to the plaintext string x and submitted to selected parallel encryption mode 61 that uses F_K, the block cipher F using the key K 31. The selected parallel encryption mode 61 has been described in Figure 3.

[0180] The parallel encryption mode 61 yields n+1 hidden ciphertext blocks z_i 87. Figure 5 shows an example where $n = 4$; i.e., the hidden ciphertext blocks 87 are z_1, z_2, z_3, z_4, z_5.

[0181] In the preferred embodiment of this invention of the stateful-sender parallel encryption, the hidden ciphertext blocks z_i 87 are submitted to a randomization step for the hidden ciphertext comprising applying a combination operation for the hidden ciphertext 84 to each hidden ciphertext block z_i 87 and each ℓ-bit element 83 of a sequence of

$n+1$ elements, resulting in $n+1$ ciphertext blocks y_i 24. The randomization step for the hidden ciphertext has been described in Figure 3. The plurality of ciphertext blocks y_i 24 forms the ciphertext string y 26 that has $n+1$ blocks. For the example presented in Figure 5, the ciphertext string 26 is $y = y_1 y_2 y_3 y_4 y_5$; i.e., has $n+1=5$ blocks. The counter ctr 72 and the ciphertext string y 26 representing the output of the encryption mode 53 form the output message data.

[0182] With the encryption of each plaintext string, the current value of the counter ctr 72 is incremented, or otherwise changed to a new value, ctr', at 73. This new value is used to encrypt the next plaintext string.

[0183] Figure 6 illustrates a schematic diagram for the preferred embodiment of this invention of the stateful-sender parallel decryption mode. From the string presented for decryption comprising the counter ctr 72 and ciphertext string y 26, the counter ctr 72 is enciphered using F_K 41, the block cipher F using key K 31, and the secret random number r_0 71 is obtained. After obtaining the secret random number r_0 71, the ciphertext string y 26, composed of $n+1$ ciphertext blocks y_i 24, is decrypted in the same manner as that used in the stateless parallel decryption mode 52 after it obtains the secret random number r_0 71 (viz., Figure 4) to obtain either the plaintext string x 23 composed of n plaintext blocks x_i 21 or the error indicator 20.

[0184] Figure 7 illustrates a schematic diagram for the preferred embodiment of this invention of the stateful parallel encryption mode. The encryption mode 55 uses a secret key K (31) and two independent secret random numbers, R 32 and R^* 33, of ℓ bits in size shared between a sender and a receiver. In the preferred embodiment of this invention, the sender and the receiver generate the same shared independent secret random numbers R 32 and R^* 33 from an already shared secret key K 31 using key separation techniques well-known in the art. In an alternate embodiment of this invention, the two independent secret random numbers, R 32 and R^* 33, are generated by a random number generator and distributed to the sender and receiver in the same way as that used for secret key K 31 using distribution techniques well-known in the art.

[0185] In this embodiment of the method of the invention a counter ctr 72 is used to obtain the block-index-independent unpredictable element $R^* \times$ ctr (modulo 2^ℓ) 74. Each block-index-independent unpredictable element 74, which is generated at the encryption of a plaintext string x 23, is unpredictable because it is obtained by combining the secret random number R^* 33 and a non-zero counter ctr 72 such that for any given ℓ-bit constant a, the probability of the event equating this element 74 and constant a is negligible, wherein the notion of negligible probability is well-known to those skilled in the art (viz., M. Naor and O. Reingold: ``From Unpredictability to Indistinguishability: A Simple Construction of Pseudo-Random Functions from MACs,'' Advances in Cryptology - CRYPTO '98 (LNCS 1462), pp. 267-282, 1998; M. Bellare, A. Desai, E. Jokipii, and P. Rogaway: ``A Concrete Security Treatment of Symmetric Encryption,'' Proceedings of the 38th Symposium on Foundations of Computer Science, IEEE, 1997, pp. 394-403). In this embodiment, each block-index-independent unpredictable element 74 is generated from the block-index-independent unpredictable element used for the encryption of the previous plaintext by modular 2^ℓ addition of the secret random number R^*, the unpredictable element used for the first encrypted plaintext being the secret random number R^* itself. In an alternate embodiment, the block-index-independent unpredictable element $R^* \times$ ctr (modulo 2^ℓ) 74 is generated by modular 2^ℓ multiplication. It should be appreciated by those skilled in the art, and is a further aspect of this invention, that the unpredictable element 74 can be generated for each plaintext to be encrypted in other ways that do not depart from the spirit and scope of the present invention as set forth in the claims.

[0186] In the preferred embodiment of this invention, the plaintext blocks x_i 21 are bit-wise exclusive-or-ed into MDC(x) 22; i.e., MDC(x) = $x_1 \oplus \ldots \oplus x_n$, and this value is appended to the plaintext string x and submitted to selected parallel encryption mode 65 that uses F_K, the block cipher F using the key K 31. The parallel encryption mode 65 comprises a plaintext randomization step applied to the n plaintext blocks x_i 21 and the MDC block 22 to generate the hidden plaintext blocks v_i 88 that are further enciphered with F_K, the block cipher F using the first key K 31, resulting in $n + 1$ hidden ciphertext blocks z_i 87. Figure 7 shows an example where $n = 4$; i.e. the hidden plaintext blocks v_i 88 are v_1, v_2, v_3, v_4, v_5 and the hidden ciphertext blocks 87 are z_1, z_2, z_3, z_4, z_5.

[0187] In the preferred embodiment of this invention of the stateful encryption, the plaintext randomization step comprises a combining each of the plaintext blocks x_i 21 and the MDC block 22, and each ℓ-bit element $E_1, E_2, \ldots, E_n$ and E^*_{n+1} 81 of a sequence of n+1 unpredictable elements for the hidden plaintext using a combination operation for the hidden plaintext 82. In the preferred embodiment of this invention, the unpredictable elements 81 are obtained as $E_i = R \times i + R^* \times ctr$ (modulo 2^{ℓ}) from the element index i for each plaintext block i, with i = 1, 2, ..., n, and as $E^*_{n+1} = R^* \times ctr$ (modulo 2^{ℓ}) for the MDC block 91. Each of these elements $E_1, E_2, \ldots, E_n$ and E^*_{n+1} 81 for the hidden plaintext is unpredictable because, for any given ℓ-bit constant a, the probability of the event $R \times i + R^* \times ctr = a$ is negligible, for i = 1, 2, ..., n, and the probability of the event $R^* \times ctr = a$ is negligible, wherein the notion of negligible probability is well-known to those skilled in the art (viz., M.

Naor and O. Reingold: ``From Unpredictability to Indistinguishability: A Simple Construction of Pseudo-Random Functions from MACs,'' Advances in Cryptology - CRYPTO '98 (LNCS 1462), pp. 267-282, 1998; M. Bellare, A. Desai, E. Jokipii, and P. Rogaway: ``A Concrete Security Treatment of Symmetric Encryption,'' Proceedings of the 38th Symposium on Foundations of Computer Science, IEEE, 1997, pp. 394-403). It should be appreciated by those skilled in the art, and is a further aspect of this invention, that the unpredictable elements 81 for the hidden plaintext and the combination operation 82 can be obtained in other ways that do not depart from the spirit and scope of the present invention as set forth in the claims. In an alternate embodiment of this invention, the unpredictable elements E_1, E_2, ..., E_n and E^*_{n+1} 81 for the hidden plaintext are obtained using the elements of the linear congruence sequence $a^i \times R$ such that $E_i = R^* \times ctr + a^i \times R$, for the n plaintext blocks and $E^*_{n+1} = R^* \times ctr$, where a is called the multiplier and is chosen to pass all the necessary spectral tests, i is the element index, $i = 1, ..., n$, and R 32 is a secret random number independent of the secret random number R* 33, as described by D.E. Knuth in ``The Art of Computer Programming - Volume 2: Seminumerical Algorithms,'' Addison-Wesley, 1981 (second edition), Chapter 3, incorporated herein by reference.

[0188] The combination operation for the hidden plaintext 82 is an operation that has an inverse. In the preferred embodiment of this invention, the combination operation 82 is the modular 2^ℓ addition, whereby each hidden plaintext block is obtained as $v_i = x_i + E_i$ modulo 2^ℓ for $i = 1, 2, ..., n$, and $v_{n+1} = x_{n+1} + E^*_{n+1}$ for the MDC block 91. In an alternate embodiment of this invention, the combination operation 82 is the bit-wise exclusive-or operation. In yet another alternate embodiment of this invention, the combination operation 82 is the modular 2^ℓ

subtraction operation. The invention, however, is not so limited, as other combination operations that have an inverse may also be used for operation for the hidden plaintext 82.

[0189] In the preferred embodiment of this invention, the distinct unpredictable elements E_i 81 (where $i \geq 1$) and the combination operation for the hidden ciphertext 82 are chosen such that for any two distinct unpredictable elements E_i, E_j, both used for the same message or each used for different messages encrypted with the same key K 31, the combination E_i op^{-1} E_j results in an ℓ-bit block that is unpredictable, where op^{-1} denotes the inverse of the combination operation. That is, for any given ℓ-bit constant a, the probability of events E_i op^{-1} $E_j = a$ and event E_i op^{-1} $E^*_{n+1} = a$ is negligible, wherein the notion of negligible probability is well-known to those skilled in the art (viz., M. Naor and O. Reingold: ``From Unpredictability to Indistinguishability: A Simple Construction of Pseudo-Random Functions from MACs,'' Advances in Cryptology - CRYPTO '98 (LNCS 1462), pp. 267-282, 1998; M. Bellare, A. Desai, E. Jokipii, and P. Rogaway: ``A Concrete Security Treatment of Symmetric Encryption,'' Proceedings of the 38th Symposium on Foundations of Computer Science, IEEE, 1997, pp. 394-403). The fact that blocks E_i op^{-1} E_j and E_i op^{-1} E^*_{n+1} are unpredictable means that enough of their ℓ bits remain unknown so that the probability of the event E_i op^{-1} Ej = a and event E_i op^{-1} $E^*_{n+1} = a$ is negligible.

[0190] In the preferred embodiment of this invention of the stateful parallel encryption, the hidden ciphertext blocks z_i 87 are submitted to a randomization step for the hidden ciphertext comprising applying a combination operation for the hidden ciphertext 84 to each hidden ciphertext block z_i 87 and each ℓ-bit element 83 of a sequence of n + 1 unpredictable elements. In the preferred embodiment of this invention, the unpredictable elements E_i 83 are obtained as $E_i = R \times i + R^* \times ctr$

(modulo 2^{ℓ}) from the element index i for each plaintext block i, with i = 1, 2, ..., n+1. Each of these elements E_i 83 is unpredictable because, for any given ℓ-bit constant a, the probability of the event $R \times i + R^* \times ctr = a$ is negligible, wherein the notion of negligible probability is well-known to those skilled in the art (viz., M. Naor and O. Reingold: ``From Unpredictability to Indistinguishability: A Simple Construction of Pseudo-Random Functions from MACs,'' Advances in Cryptology - CRYPTO '98 (LNCS 1462), pp. 267-282, 1998; M. Bellare, A. Desai, E. Jokipii, and P. Rogaway: ``A Concrete Security Treatment of Symmetric Encryption,'' Proceedings of the 38th Symposium on Foundations of Computer Science, IEEE, 1997, pp. 394-403). It should be appreciated by those skilled in the art, and is a further aspect of this invention, that the unpredictable elements for the hidden ciphertext E_i 83 and the combination operation for the hidden ciphertext 84 can be obtained in other ways that do not depart from the spirit and scope of the present invention as set forth in the claims. In an alternate embodiment of this invention, the unpredictable elements E_i 83 for the hidden ciphertext are obtained using the elements of the linear congruence sequence $a^i \times R$ such that $E_i = R^* \times ctr + a^i \times R$, where a is called the multiplier and is chosen to pass all the necessary spectral tests, i is the element index, i = 1, ..., n+1, and R 32 is a secret random number independent of the secret random number R* 33, as described by D.E. Knuth in ``The Art of Computer Programming - Volume 2: Seminumerical Algorithms,'' Addison-Wesley, 1981 (second edition), Chapter 3, incorporated herein by reference.

[0191] The combination operation for the hidden ciphertext 84 is an operation that has an inverse. In the preferred embodiment of this invention, the combination operation 84 is the modular 2^{ℓ} addition, whereby each ciphertext block is obtained as $y_i = z_i + E_i$ modulo 2^{ℓ}. In an alternate embodiment of this invention, the combination operation 84 is the bit-wise exclusive-or operation. In yet another alternate embodiment of this invention, the combination operation 84 is the modular 2^{ℓ} subtraction operation. The invention, however, is not so limited, as other combination operations that have an inverse may also be used for operation for the hidden ciphertext 84.

[0192] In the preferred embodiment of this invention, the distinct unpredictable elements E_i 83 (where $i \geq 1$) and the combination operation for the hidden ciphertext 84 are chosen such that for any two distinct unpredictable elements E_i, E_j, both used for the same message or each used for different messages encrypted with the same key K 31, the combination E_i op^{-1} E_j results in an ℓ-bit block that is unpredictable, where op^{-1} denotes the inverse of the combination operation. That is, for any given ℓ-bit constant a, the probability of the event E_i op^{-1} $E_j = a$ is negligible, wherein the notion of negligible probability is well-known to those skilled in the art (viz., M. Naor and O. Reingold: ``From Unpredictability to Indistinguishability: A Simple Construction of Pseudo-Random Functions from MACs,'' Advances in Cryptology - CRYPTO '98 (LNCS 1462), pp. 267-282, 1998; M. Bellare, A. Desai, E. Jokipii, and P. Rogaway: ``A Concrete Security Treatment of Symmetric Encryption,'' Proceedings of the 38th Symposium on Foundations of Computer Science, IEEE, 1997, pp. 394-403). The fact that block E_i op^{-1} E_j is unpredictable means that enough of its ℓ bits remain unknown so that the probability of the event E_i op^{-1} $Ej = a$ is negligible.

[0193] The application of the combination operation for the hidden ciphertext 84 to the plurality of hidden ciphertext blocks 87 and the unpredictable elements for the hidden ciphertext 83 of the sequence results in a plurality of ciphertext blocks y_i 24. The plurality of ciphertext blocks y_i 24 forms the ciphertext string y 26 that has n+1 blocks. For the example presented in Figure 7, the ciphertext string 26 is $y = y_1\ y_2\ y_3\ y_4\ y_5$; i.e., has $n+1=5$ blocks. The counter ctr 72 and the ciphertext string y 26 representing the output of the encryption mode 55 form the output message data.

[0194] With the encryption of each plaintext string, the current value of the counter ctr 72 is incremented, or otherwise changed, to a new non-zero value, ctr′, at 73. This new value is used to encrypt the next plaintext string.

[0195] Figure 8 illustrates a schematic diagram for the preferred embodiment of this invention of the stateful parallel decryption mode. The decryption mode 56 uses a secret key K (31) and two independent random numbers, R 32 and R^* 33, shared between a sender and a receiver. The string presented for decryption comprises the non-zero counter ctr 72 and ciphertext string y 26. In this embodiment of the method of the invention, a non-zero counter ctr 72 is used to obtain the unpredictable element $R^* \times ctr$ (modulo 2^{ℓ}) 74 in the same way as at encryption (viz., Figure 7). The secret shared random numbers R 32 and R^* 33 are used to obtain the unpredictable elements for the hidden ciphertext $E_i = R \times i + R^* \times ctr$ (modulo 2^{ℓ}) 83 in the same way as at encryption (viz., Figure 7). These unpredictable elements E_i 83 and the ciphertext blocks y_i 24 are combined using the inverse combination operation for the ciphertext 85 to generate the hidden ciphertext blocks z_i 87. The inverse combination operation for the hidden ciphertext 85 is the inverse of the combination operation for the hidden ciphertext 84 used at encryption. In the preferred embodiment of this invention of the stateful parallel decryption, the inverse combination operation for the ciphertext

85 is subtraction modulo 2^{ℓ}; i.e., $z_i = y_i - (R \times i + R^* \times ctr)$. In an alternate embodiment of this invention, when the combination operation 84 is the bit-wise exclusive-or operation, the inverse combination operation for the ciphertext 85 is the bit-wise exclusive-or operation; i.e., $z_i = y_i \oplus (R \times i + R^* \times ctr)$. In another alternate embodiment of this invention, when the combination operation 84 is the modular 2^{ℓ} subtraction operation, the inverse combination operation for the ciphertext 85 is addition modulo 2^{ℓ}; i.e., $z_i = y_i + (R \times i + R^* \times ctr)$. The invention, however, is not so limited, as other inverse combination operations may also be used for operation 85, the only restriction being that operation 85 is the inverse of the combination operation for the hidden ciphertext 84.

[0196] The n+1 hidden ciphertext blocks z_i 87 are presented to the select parallel decryption mode 66 that uses F^{-1}_K, the inverse of the block cipher F using key K 31. The parallel decryption mode 66 comprises deciphering the n+1 hidden ciphertext blocks z_i 87 using F^{-1}_K, the inverse of the block cipher F using key K 31 to obtain n+1 hidden plaintext blocks v_i 88 that are further submitted to a reverse plaintext randomization step that generates n+1 blocks x_i. The last block x_{n+1} 29 represents the decrypted MDC block.

[0197] The reverse plaintext randomization step comprises applying the inverse operation for the hidden plaintext 86 to the n+1 hidden plaintext blocks v_i 88 and the n+1 unpredictable elements for the hidden plaintext E_1, E_2, ..., E_n and E^*_{n+1} 81 obtained in the same way as at encryption (viz., Figure 7). The inverse combination operation for the hidden plaintext 86 is the inverse of the combination operation for the hidden plaintext 82 used at encryption. In the preferred embodiment of this invention of the stateless parallel decryption, the inverse combination operation for the plaintext 86 is subtraction modulo 2^{ℓ}; i.e., $x_i = v_i - (R \times i + R^* \times ctr)$, for $1 \leq i \leq n$, and $x_{n+1} = v_{n+1} - (R^* \times ctr)$ for $i = n + 1$. In an alternate embodiment of this invention, when the combination operation 82 is the bit-wise exclusive-or operation, the inverse combination operation for the hidden plaintext 85 is the bit-wise exclusive-or operation; i.e., $x_i = v_i \oplus (R \times i + R^* \times ctr)$, for $1 \leq i \leq n$, and $x_{n+1} = v_{n+1} \oplus (R^* \times ctr)$ for $i = n + 1$. In another alternate embodiment of this invention, when the combination operation 82 is the modular 2^{ℓ} subtraction operation, the inverse combination operation for the hidden plaintext 86 is addition modulo 2^{ℓ}; i.e., $x_i = v_i + (R \times i + R^* \times ctr)$, for $1 \leq i \leq n$, and $x_{n+1} = v_{n+1} + (R^* \times ctr)$ for $i = n + 1$. The invention, however, is not so limited, as other inverse combination operations may also be used for operation 86, the only restriction being that operation 86 is the inverse of the combination operation for the hidden plaintext 82.

[0198] The n blocks x_i, namely $x_1, x_2, \ldots, x_n$, in accordance with one embodiment of the MDC function, are bit-wise exclusive-or-ed to obtain computed MDC(x) block 91; i.e. $MDC(x) = x_1 \oplus \ldots \oplus x_n$. Then the computed MDC(x) and the decrypted MDC block x_{n+1} 29 are compared for equality at 92. If the computed MDC block MDC(x) 91 and the decrypted MDC block 29 are not equal then the result of the decryption of the data string y 26 is the error indicator 20. If the computed MDC block MDC(x) 91 and the decrypted MDC block 29 are equal then the output from the logical "and" operators 93 is the result of the decryption of the ciphertext string y 26 using the decryption mode 56; i.e., the result is the plaintext string x 23 composed of n plaintext blocks x_i 21. For the example illustrated in Figure 8, the output of the decryption mode 56 is the plaintext string 23 $x = x_1\ x_2\ x_3\ x_4$.

[0199] Figure 9 illustrates a schematic diagram for the preferred embodiment of the L-segment stateful-sender parallel encryption mode. Input plaintext string x 23 composed of n plaintext blocks x_i 21 is encrypted using a secret key K 31 to obtain output ciphertext string y 26 composed of ciphertext blocks y_i 24. The plaintext string x 23 (which is padded in a standard way as necessary) is partitioned into a plurality of plaintext segments 27. Each plaintext segment contains a plurality of plaintext blocks x_i 21. Figure 9 shows an example in which the number of segments is L=3, and the plaintext string x 23 has 12 plaintext blocks x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} x_{11} x_{12}; furthermore, plaintext segment 1 is composed of plaintext blocks x_1 x_2 x_3 x_4, plaintext segment 2 is composed of plaintext blocks x_5 x_6 x_7 x_8, and plaintext segment 3 is composed of plaintext blocks x_9 x_{10} x_{11} x_{12}. Note that although in the example presented in Figure 9, the plaintext segments 27 have the same number of plaintext blocks 21, this is not required.

[0200] For each plaintext segment 27, a variant 75 of the counter ctr 72 is enciphered using F_K 41, the block cipher F using the secret key K 31, to yield the per-segment secret random numbers r_{0i} 71. Figure 9 shows an example in which the per-segment variant 75 of the counter is computed from the counter by adding modulo 2^{ℓ}, a segment index, i.e., for plaintext segment 1 use ctr as the variant of the counter 75 and compute the first segment random number r_{01} 71 as $r_{01} = F_K(ctr)$, for plaintext segment 2 use ctr + 1 modulo 2^{ℓ} as the variant of the counter 75 and compute the second segment random number r_{02} 71 as $r_{02} = F_K(ctr + 1)$, and for plaintext segment 3 use ctr + 2 modulo 2^{ℓ} as the variant of the counter 75 and compute the third segment random number r_{03} 71 as $r_{03} = F_K(ctr + 2)$. Each per-segment random number 71 and the plaintext segment 27 are submitted to a stateful-sender parallel encryption mode 53 (e.g., Figure 5) using the secret key K 31 that generates the ciphertext blocks 24 of output ciphertext segment 28. The ciphertext segments 28 are further assembled together with the number of ciphertext segments L, the length of each ciphertext segment and the ciphertext segment sequence into the ciphertext string y 26 (e.g., by standard ASN.1 encoding). The ciphertext string y 26 contains n + L ciphertext blocks. Figure 9 shows an example in which plaintext segment 1 is encrypted using the parallel encryption mode 53, the secret random number r_{01} generated at 71, the secret key K 31 to obtain the ciphertext blocks y_1 y_2 y_3 y_4 y'_5; plaintext segment 2 is encrypted using the parallel encryption mode 53, the secret random number r_{02} generated at 71, the secret key K 31 to obtain the ciphertext blocks y_5 y_6 y_7 y_8 y'_9; and plaintext segment 3 is encrypted using the parallel encryption mode 53, the secret random number r_{03} generated at 71, the secret key K 31 to obtain the ciphertext blocks y_9 y_{10} y_{11} y_{12} y'_{13}. In the example presented in Figure 9, the ciphertext string 26 is $y = y_1\ y_2\ y_3\ y_4\ y'_5\ y_5\ y_6\ y_7\ y_8\ y'_9\ y_9\ y_{10}\ y_{11}\ y_{12}\ y'_{13}$ and contains $n + L = 12 + 3 = 15$ ciphertext blocks.

[0201] With the encryption of each plaintext string, the current value of the counter ctr is incremented with the number of plaintext segments L, or otherwise changed to a new value, at 73. This new value is used to encrypt the next plaintext string.

[0202] Figure 10 illustrates a schematic diagram for the preferred embodiment of the L-segment stateful-sender parallel decryption mode. Input ciphertext string y 26 is decrypted at 54 to obtain a plurality of output plaintext segments x 27 or failure indicators 20. The parsing of the string encoding of y 26 yields the ctr 72, the number of ciphertext segments L, the length of each ciphertext segment and the ciphertext segment sequence. The parsing of the ciphertext string y yields the number of ciphertext segments L, the length of each ciphertext segment and the ciphertext segment sequence; furthermore, the ciphertext string y 26 is partitioned into a plurality of ciphertext segments 28. Each segment contains a plurality of ciphertext blocks y_i 24. Figure 10 shows an example in which the number of segments is L = 3, the ciphertext string y 26 has 15 ciphertext blocks y_1 y_2 y_3 y_4 y'_5 y_5 y_6 y_7 y_8 y'_9 y_9 y_{10} y_{11} y_{12} y'_{13}; furthermore, ciphertext segment 1 is composed of ciphertext blocks y_1 y_2 y_3 y_4 y'_5, ciphertext segment 2 is composed of ciphertext blocks y_5 y_6 y_7 y_8 y'_9, and ciphertext segment 3 is composed of ciphertext blocks y_9 y_{10} y_{11} y_{12} y'_{13}. Note that although in the example presented in Figure 10, the ciphertext segments 28 have the same number of ciphertext blocks 24, this is not required.

[0203] From the counter ctr 72, the per-segment secret random number r_{0i} 71 are obtained in the same manner as at a segmented encryption mode. For each ciphertext segment 28, a variant 75 of the counter ctr 72 is enciphered using F_K 41, the block cipher F using a secret key K 31, to yield the per-segment secret random numbers r_{0i} 71. Figure 10 shows an example in which the per-segment variant 75 of the counter is computed from the counter by adding modulo 2^ℓ, a segment index, i.e., for

ciphertext segment 1 use ctr as the variant of the counter 75 and compute the first segment random number r_{01} 71 as $r_{01} = F_K(ctr)$, for ciphertext segment 2 use ctr + 1 modulo 2^ℓ as the variant of the counter 75 and compute the second segment random number r_{02} 71 as $r_{02} = F_K(ctr + 1)$, and for ciphertext segment 3 use ctr + 2 modulo 2^ℓ as the variant of the counter 75 and compute the third segment random number r_{03} 71 as $r_{03} = F_K(ctr + 2)$. Each per-segment random number 71 and the ciphertext segment 28 are submitted to a stateful-sender parallel decryption mode 54 (viz., Figure 6) using the secret key K 31 that generates the plaintext blocks 21 of output plaintext segment 27 or the failure indicator 20.

[0204] Each plaintext segment 27 is either accepted, or it is rejected if the output of the stateful-sender parallel decryption mode 54 is the failure indicator 20.

[0205] Figure 11 illustrates a schematic diagram for the preferred embodiment of the L-segment stateful parallel encryption mode. Input plaintext string x 23 composed of n plaintext blocks x_i 21 is encrypted using a secret key K 31 to obtain output ciphertext string y 26 composed of ciphertext blocks y_i 24. The plaintext string x 23 (which is padded in a standard way as necessary) is partitioned into a plurality of plaintext segments 27. Each plaintext segment contains a plurality of plaintext blocks x_i 21. Figure 11 shows an example in which the number of segments is L=3, and the plaintext string x 23 has 12 plaintext blocks x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} x_{11} x_{12}; furthermore, plaintext segment 1 is composed of plaintext blocks x_1 x_2 x_3 x_4, plaintext segment 2 is composed of plaintext blocks x_5 x_6 x_7 x_8, and plaintext segment 3 is composed of plaintext blocks x_9 x_{10} x_{11} x_{12}. Note that although in the example presented in Figure 11, the plaintext segments 27 have the same number of plaintext blocks 21, this is not required.

[0206] For each plaintext segment 27, a per-segment unpredictable element is created at 74 from a first secret random number R^* 33 and the non-zero counter 72; i.e., for plaintext segment 1, the per-segment unpredictable element 74 is $R^* \times \text{ctr}$ (modulo 2^{ℓ}), for plaintext segment 2, the per-segment unpredictable element 74 is $R^* \times (\text{ctr} + 1)$ (modulo 2^{ℓ}), for plaintext segment 3, the per-segment unpredictable element 74 is $R^* \times (\text{ctr} + 2)$ (modulo 2^{ℓ}).

[0207] Each per-segment unpredictable element 74 and the plaintext segment 27 are submitted to a stateful parallel encryption mode 55 (viz., Figure 7) using the secret key K 31 that generates the ciphertext blocks 24 of output ciphertext segment 28. The ciphertext segments 28 are further assembled together with the number of ciphertext segments L, the length of each ciphertext segment and the ciphertext segment sequence into the ciphertext string y 26 (e.g., by standard ASN.1 encoding). The ciphertext string y 26 contains n + L ciphertext blocks. Figure 11 shows an example in which plaintext segment 1 is encrypted using the parallel encryption mode 55, the per-segment unpredicatable element $R^{*} \times$ ctr (modulo 2^{ℓ}) generated at 74, the secret key K 31 to obtain the ciphertext blocks $y_1\ y_2\ y_3\ y_4\ y'_5$; plaintext segment 2 is encrypted using the parallel encryption mode 55, the per-segment unpredictable element $R^{*} \times$ (ctr + 1) (modulo 2^{ℓ}) generated at 74, the secret key K 31 to obtain the ciphertext blocks $y_5\ y_6\ y_7\ y_8\ y'_9$; and plaintext segment 3 is encrypted using the parallel encryption mode 55, the per-segment unpredictable element $R^{*} \times$ (ctr + 2) (modulo 2^{ℓ}) generated at 74, the secret key K 31 to obtain the ciphertext blocks $y_9\ y_{10}\ y_{11}\ y_{12}\ y'_{13}$. In the example presented in Figure 11, the ciphertext string 26 is $y = y_1\ y_2\ y_3\ y_4\ y'_5\ y_5\ y_6\ y_7\ y_8\ y'_9\ y_9\ y_{10}\ y_{11}\ y_{12}\ y'_{13}$ and contains n + L = 12 + 3 = 15 ciphertext blocks.

[0208] With the encryption of each plaintext string, the current value of the non-zero counter ctr is incremented with the number of plaintext segments L, or otherwise changed to a new non-zero value, at 73. This new value is used to encrypt the next plaintext string.

[0209] Figure 12 illustrates a schematic diagram for the preferred embodiment of the L-segment stateful parallel decryption mode. Input ciphertext string y 26 is decrypted at 56 to obtain a plurality of output plaintext segments x 27 or failure indicators 20. The parsing of the string encoding of y 26 yields the ctr 72, the number of ciphertext segments L, the length of each ciphertext segment and the ciphertext segment sequence. The parsing of the ciphertext string y yields the number of ciphertext segments L, the length of each ciphertext segment and the ciphertext segment sequence; furthermore, the ciphertext string y 26 is partitioned into a plurality of ciphertext segments 28. Each segment contains a plurality of ciphertext blocks y_i 24. Figure 12 shows an example in which the number of segments is L = 3, the ciphertext string y 26 has 15 ciphertext blocks y_1 y_2 y_3 y_4 y'_5 y_5 y_6 y_7 y_8 y'_9 y_9 y_{10} y_{11} y_{12} y'_{13}; furthermore, ciphertext segment 1 is composed of ciphertext blocks y_1 y_2 y_3 y_4 y'_5, ciphertext segment 2 is composed of ciphertext blocks y_5 y_6 y_7 y_8 y'_9, and ciphertext segment 3 is composed of ciphertext blocks y_9 y_{10} y_{11} y_{12} y'_{13}. Note that although in the example presented in Figure 12, the ciphertext segments 28 have the same number of ciphertext blocks 24, this is not required.

[0210] From the non-zero counter ctr 72, the per-segment secret unpredictable elements 74 are obtained in the same manner as at a segmented encryption mode; i.e., for ciphertext segment 1, the per-segment unpredictable element 74 is $R^* \times$ ctr (modulo 2^{ℓ}), for ciphertext segment 2, the per-segment unpredictable element 74 is $R^* \times$ (ctr + 1) (modulo 2^{ℓ}), for ciphertext segment 3, the per-segment unpredictable element 74 is $R^* \times$ (ctr + 2) (modulo 2^{ℓ}).

[0211] Each per-segment unpredictable element 74 and the ciphertext segment 28 are submitted to a stateful parallel decryption mode 56 (e.g., Figure 8) using the secret key K 31 that generates the plaintext blocks 21 of output plaintext segment 27 or the failure indicator 20.

[0212] Each plaintext block 27 is either accepted, or it is rejected if the output of the stateful parallel decryption mode 56 is the failure indicator 20.

[0213] It is readily understood by those skilled in the art that similar modes can be derived for stateless segmented encryption method and stateless decryption method, wherein, in the preferred embodiment, the per-segment random numbers r_{0i} 71 are generated by a random number generator. In an alternate embodiment, the per-segment random numbers r_{0i} 71 are generated from the shared secret key K 31 by key-separation techniques well-known in the art.

[0214] Additional properties of the method of this invention are now presented. In a further aspect, the method of this invention allows the incremental replacement of ciphertext blocks without requiring the complete re-execution of the decryption and encryption procedure. That is, if a plaintext block x_i of an n-block encrypted string x needs to be updated to obtain new plaintext block x'_i of new string x', then the ciphertext block y_i of the i-th block ciphertext string y is replaced with a new block y'_i. A new MDC(x') block and ciphertext blocks y'_i and y'_{n+1} are computed using only a small number of invocations of the block cipher that does not depend on the number of blocks of the input plaintext string x and of the ciphertext string y of the original . For instance, for the preferred embodiment of the stateless parallel encryption mode using secret key K, if R* and R (viz., Figures 7), are the random independent secret ℓ-bit numbers used in the encryption of the original input plaintext string x, then the ciphertext string y' in which block y_i is replaced with a new block y'_i, representing the enciphering of updated plaintext block x'_i, then the ciphertext y' of plaintext string x' is thus computed as follows.

[0215] The new block x'_i is used to update original plaintext block x_{n+1} $= MDC(x) = x_1 \oplus \ldots \oplus x_n$ and obtain plaintext block $x'_{n+1} = MDC(x') =$

$MDC(x) \oplus x'_i \oplus x_i$. The new blocks x'_i and x'_{n+1} are used to generate two new ciphertext blocks y'_i and y'_{n+1}. Both ciphertext blocks y'_i and y'_{n+1} are generated using the steps defined in Figure 7. To obtain new ciphertext y'_i block x'_i is subjected to a randomization step comprising, in one embodiment, applying a combination operation 82 (viz., Figure 7) with the i-th element E_i of a sequence of n + 1 unpredictable ℓ-bit elements 81. The resulting ℓ-bit hidden plaintext block v'_i 88 is enciphered with block cipher F_K 41 using secret key K 31 to obtain the hidden ciphertext block z'_i 87. This hidden ciphertext block is further randomized by applying a combination operation 84 (viz., Figure 7) with the i-th element E_i (viz., Figure 7) to obtain the desired ciphertext y'_i. To obtain new ciphertext y'_{i+1}, block x'_{i+1} is subjected to a randomization step comprising, in one embodiment, applying a combination operation 82 (viz., Figure 7) with the n + 1-st element E^*_{n+1} of a sequence of n + 1 unpredictable ℓ-bit elements 81. The resulting ℓ-bit hidden plaintext block v_{n+1} 88 is enciphered with block cipher F_K 41 using secret key K 31 to obtain the hidden ciphertext block z_{n+1} 87. This hidden ciphertext block is further randomized by applying a combination operation 84 (viz., Figure 7) with the n + 1-st element E_{n+1} (viz., Figure 7) to obtain the desired ciphertext y'_{n+1}.

[0216] It is readily understood by those skilled in the art that deletion or insertion of a ciphertext block y'_i , $2 \leq i \leq n$, can also be performed without requiring the complete execution of the message decryption and encryption procedures. Furthermore, it is also readily understood by those skilled in the art that the incremental replacement, deletion, or insertion of a plurality of ciphertext blocks without requiring the complete execution of the message decryption and encryption procedures applies to all other embodiments of this invention, not just to the parallel stateful encryption mode described at Figures 7 and 8.

[0217] In a yet further aspect of this invention, the method of this invention allows out-of-order processing of both plaintext and ciphertext blocks of a message. Referring to the preferred embodiment of the stateful parallel decryption mode using secret key K 31 (viz., Figure 8), if any ciphertext block y_i is received before the other ciphertext blocks, then the corresponding unpredictable element for the hidden ciphertext E_i 83 and the corresponding unpredictable element for the hidden plaintext E_i for $1 \leq i \leq n$ and E^*_{n+1} for $i = n + 1$ 81 can be computed immediately, and the inverse combination operation for the hidden ciphertext 85 and the inverse combination operation for the hidden plaintext 86 can be performed immediately; i.e., there is no delay for any additional deciphering or enciphering operation. Also, for the preferred embodiment of the parallel stateful encryption mode using secret key K 31 (viz., Figure 7), if any plaintext block x_i is received before the other plaintext blocks, then the corresponding unpredictable element for the hidden plaintext E_i for $1 \leq i \leq n$ and E^*_{n+1} for $i = n + 1$ 81 and the corresponding unpredictable element for the hidden ciphertext E_i 83 can be computed immediately, and the combination operation for the hidden plaintext 82 and the combination operation for the hidden ciphertext 84 can be performed immediately; i.e., there is no delay for any additional deciphering or enciphering operation.

[0218] It is readily understood by those skilled in the art that the out-of-order processing of applies to all other embodiments of this invention, not just to the parallel stateful encryption mode using secret key K 31(described in Figures 7 and 8).

[0219] Additional details of the embodiment of the method of the present invention are now presented. The encryption modes presented in this method processes plaintext strings whether or not they are multiple of a desired block length ℓ. The method begins by selecting F, an ℓ-bit block cipher using keys of length k. For example, ℓ is 64 and k = 56 when F is the DES algorithm. Of course, other block ciphers are known to those skilled in the art, and some of these block ciphers have been surveyed by Menezes, Van Oorschot and Vanstone in their book entitled "Handbook of Applied Cryptography," CRC Press, 1997 hereby included by reference.

[0220] In the preferred embodiments of the stateless mode and of the stateful-sender mode, padding the plaintext string 23 comprises the following steps: if the last block x_n of the plaintext has ℓ bits in length derive a last element E^*_{n+1} of the sequence of unpredictable elements for the hidden plaintext 81 to be combined with the MDC block 22 (i.e., block x_{n+1}) from the bit-wise complement s_0 of a random number r_0 71, namely $E^*_{n+1} = s_0 \times (n+2)$ modulo 2^ℓ; else, append to the last block of the plaintext x_n the bit 1 and the necessary bits of 0 to generate a last equal block 21, and derive a last element E^*_{n+1} of the sequence of unpredictable elements for the hidden plaintext 81 to be combined with the MDC block 22 (i.e., block x_{n+1}) from the random number r_0 71, namely $E^*_{n+1} = r_0 \times (n+2)$ modulo 2^ℓ. In these preferred embodiments of the stateless mode and of the stateful-sender mode, each but the last of the plurality of the unpredictable elements (81) of the sequence of unpredictable elements for the hidden plaintext is generated by combining a different element identifier i for each of the unpredictable elements and the secret random number r_0 71; i.e., $E_i = r_0 \times i$ modulo 2^ℓ for plaintext blocks with i = 1, 2, ..., n. In the preferred embodiment of the stateful mode, padding the plaintext string 23 consists of the following steps: if the last block x_n of the plaintext has ℓ bits in length derive a last element E^*_{n+1} of the sequence of unpredictable elements for the hidden plaintext 81 to be combined with the MDC block 22 (i.e., block x_{n+1}) from the bit-wise complement S^* of random number R^* 33 , namely $E^*_{n+1} = S^* \times \text{ctr} \bmod 2^\ell$; else, append to the last block of the plaintext x_n the bit 1 and the necessary bits of 0 to generate a last equal block 21, and derive a last element E^*_{n+1} of the sequence of unpredictable elements for the hidden plaintext 81 to be combined with the MDC block 22 (i.e., block x_{n+1}) from the random number R^* 33, namely $E^*_{n+1} = R^* \times \text{ctr} \bmod 2^\ell$. In this preferred embodiment of the stateful mode, each but the last of the plurality of the unpredictable elements (81) of the sequence of unpredictable elements for the hidden plaintext is generated as: $E_i = R \times i + R^* \times \text{ctr} \bmod 2^\ell$ for plaintext blocks with $i = 1, 2, \ldots, n$. In an alternate embodiment, the input plaintext string x 23 is padded in some standard fashion as necessary so that it is a multiple of ℓ bits. In this alternate embodiment, the padding is commonly known in the data processing art.

[0221] It should be appreciated by those skilled in the art that the specific embodiments disclosed above may be readily utilized as a basis for modifying or designing other techniques and routines for carrying out the same purposes and spirit of the present invention as set forth in the claims.

[0222] The foregoing description of a preferred embodiment of the invention has been presented for purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed, and modifications and variations are possible in light of the above teachings or may be acquired from practice of the invention. The embodiments were chosen and described in order to explain the principles of the invention and its practical application to enable one skilled in the art to utilize the invention in various embodiments and with various modifications as are suited to the particular use contemplated. It is intended that the scope of the invention be defined the claims appended hereto, and their equivalents.